On the Low-density Latent Regions of VAE-based Language Models

Ruizhe Li* R.LI@SHEF.AC.UK
Xutan Peng* X.PENG@SHEF.AC.UK
Chenghua Lin† C.LIN@SHEF.AC.UK
Department of Computer Science, The University of Sheffield, UK
Wenge Rong W.RONG@BUAA.EDU.CN
School of Computer Science and Engineering, Beihang University, China
Zhigang Chen CHENZHIGANG@ZWU.EDU.CN
College of Information and Intelligent Engineering, Zhejiang Wanli University, China

Abstract

By representing semantics in latent spaces, Variational autoencoders (VAEs) have been proven powerful in modelling and generating signals such as image and text, even without supervision. However, previous studies suggest that in a learned latent space, some low-density regions (aka. *holes*) exist, which could harm the overall system performance. While existing studies focus on empirically mitigating these latent holes, how they distribute and how they affect different components of a VAE, are still unexplored. In addition, the hole issue in VAEs for language processing is rarely addressed. In our work, by introducing a simple hole-detection algorithm based on the neighbour consistency between VAE's input, latent, and output semantic spaces, we propose to deeply dive into these topics for the first time. Comprehensive experiments including automatic evaluation and human evaluation imply that large-scale low-density latent holes may not exist in the latent space. In addition, various sentence encoding strategies are explored and the native word embedding is the most suitable strategy for VAEs in language modelling task.

Keywords: Variational Autoencoder, Low-density Regions, Latent Holes

1. Introduction

The Variational autoencoder (VAE) (Kingma and Welling, 2013; Rezende et al., 2014) is a powerful model to unsupervisedly learn a low-dimensional manifold (aka. a latent space) from a non-trivial high-dimensional data manifold. It has been proven useful in multiple downstream applications: the encoder of a VAE can facilitate multiple tasks such as classification (Xu et al., 2017) and transfer learning (Higgins et al., 2017b), while the decoder holds promise in the generation domain (Fang et al., 2019; He et al., 2019).

Despite its success in processing image (Huang et al., 2018; Razavi et al., 2019; Li et al., 2020c), text (Bowman et al., 2016; He et al., 2019; Li et al., 2019b, 2020b) and audio (Roberts et al., 2018), past studies report that a sampled latent variable might land in low-density

* Equal contribution.
† Corresponding author.

regions (aka. *holes*) of the learned latent space (Rezende and Viola, 2018; Xu et al., 2019). Existing approaches concentrate on directly mitigating the hole problem in an empirical fashion, and mainly focus on the image domain. Falorsi et al. (2018) proposed to use the manifold-valued latent variables to learn a latent space; Davidson et al. (2018) introduced the von Mises-Fisher (vMF) distribution to replace the conventional Gaussian distribution; Kalatzis et al. (2020) proposed to use the Riemannian Brownian motion prior rather than the simple Gaussian prior. In the text field, the existence of latent holes has just been confirmed by Xu et al. (2019) very recently, who additionally claimed the "holes problem" tends to be more severe on text compared with image. They proposed to constrain the latent variable to an orthogonal and no-holes filled probability simplex and manipulate the latent code within the simplex for text style transfer.

To summarise, all these studies simply attempt to alleviate holes by constructing a theoretically less-hole space or replacing the prior. They failed to identify the locations of these holes, to investigate how they *respectively* affect the trained encoder and decoder, or to reveal their (semantic) properties. Also, the research on holes of VAEs for text is relatively neglected and is still at an initial stage.

In this work, we propose the first fine-grained framework to automatically detect low-density latent regions of VAEs, with a focus on the natural language processing scenario. Our algorithm is based on the consistency of neighbouring representation spaces for the inputs, outputs and latent variables, which is derived from a popular semantic transferring experiment (i.e., the latent variable linear interpolation) in texts (Bowman et al., 2016; Zhao et al., 2018; Fang et al., 2019; Shen et al., 2019; Li et al., 2020a). Moreover, our method can separately analyses the holes' influence on the performance of the encoder and decoder: we believe this direction has never been visited.

To validate the effectiveness of our algorithm and to get more insights on the holes' properties, we design three comprehensive experiments. Firstly, we evaluate different sentence encoding schemes to find out the best way to represent VAEs' input and output semantics spaces, so as to guarantee the accuracy of our method (§ 4.2). Secondly, we detect holes in the latent space and examine how they affect encoders and decoders respectively, through automatic and human evaluations (§ 4.3). Thirdly, we further investigate whether the identified holes really encode nothing at all as past studies hypothesised (Rezende and Viola, 2018; Xu et al., 2019), or they actually capture information which is yet to be explored based on the identified holes in the second experiment (§ 4.4). Comprehensive experiments show that the native word embedding is the best way to encode sentences for VAEs in language modelling task. However, the negative automatic and human evaluation results imply that the large-scale low-density holes may not exist in the latent space. The experiment which investigates whether holes are really vacant or not can not be validated because of the negative results of large latent holes hypothesis.

2. Background: Variational Autoencoder

A variational autoencoder is a generative model which defines a joint distribution over the observations $\mathbf{x}$ and the latent variables $\mathbf{z}$, i.e., $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$. Given a dataset $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ with N i.i.d. datapoint, we need to optimise the marginal likelihood $\frac{1}{N}p(\mathbf{X}) = \frac{1}{N}\sum_i^N \int p(\mathbf{x}_i|\mathbf{z}_i)p(\mathbf{z}_i)d\mathbf{z}$ over the entire training set. However, this marginal likelihood is

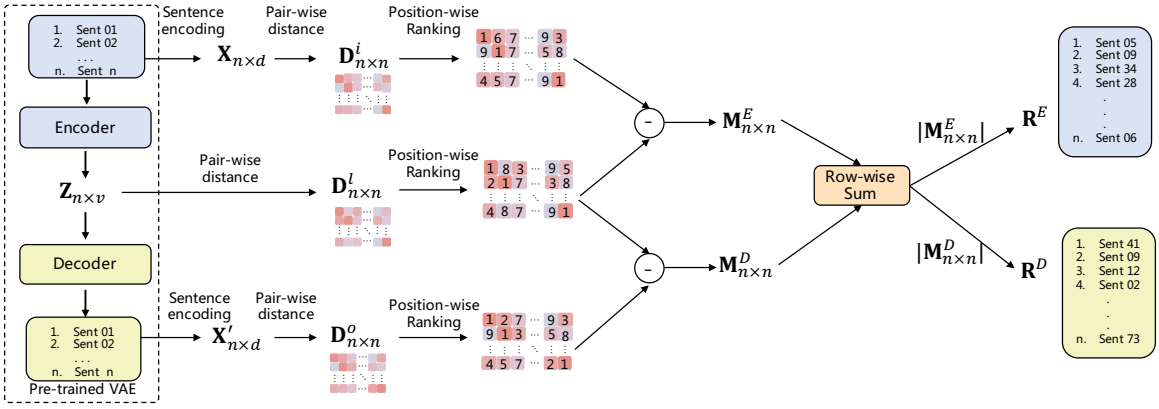


Figure 1: The framework of our methodology.

intractable. The common solution for this issue is to maximise the *Evidence Lower Bound* (ELBO) using the variational inference for every observation $\mathbf{x}$:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})] - D_{\mathrm{KL}}\left(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z})\right) , \tag{1}$$

where $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ is a variational posterior to approximate the true posterior $p(\mathbf{z}|\mathbf{x})$. Both the variational posterior $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ (aka. encoder) and the conditional distribution $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ (aka. decoder) are set up using two neural networks with parameters $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$, respectively. Normally, the first term in Eq. (1) is the expected data reconstruction loss demonstrating how well the model can reconstruct data given a latent variable. The second term is the KL-divergence of the approximate variational posterior from the prior, i.e., a regularisation forcing the learned posterior to be as close to the prior as possible.

3. Methodology

For each of n samples in a test set, our framework aims to detect how likely it is to link with a latent hole, and whether it has influence on the performance of the encoder, the decoder or both.

As sketched in Fig. 1, the main pipeline begins with the testing inputs and outputs of a pre-trained VAE-based language model, where the representation matrices are denoted as $\mathbf{X}$ and $\mathbf{X}'$, respectively (they all have n rows which correspond to n sentences with d dimensions; see § 4.2 for implementation choices). When researchers build VAE-based language models, one commonly adopted hypothesis is that the sentence encoding of inputs, outputs, and latent variables are all semantically smooth (Bowman et al., 2016; Zhao et al., 2018; Fang et al., 2019; Shen et al., 2019; Li et al., 2020a). This belief has been evidenced by the popular semantic transferring experiments from sentence $\mathbf{x}_1 \in \mathbf{X}$ to $\mathbf{x}_2 \in \mathbf{X}$ with linear interpolation between the corresponding $\mathbf{z}_1 \in \mathbf{Z}$ and $\mathbf{z}_2 \in \mathbf{Z}$, where $\mathbf{Z}$ denotes the latent variables matrix which row-wise corresponds to $\mathbf{X}$ and $\mathbf{X}'$ (i.e., with n rows) and has a dimension of v. Therefore, suppose a *perfect* VAE which is hole-free, then for each sample, its neighbouring structures in the input, latent, and output spaces should be consistent,

Table 1: Statistics of the Yelp 2015, Yahoo, SNLI datasets.

Dataset	Train	Dev.	Test	Avg. length	Vocab.
Yelp15	100,000	10,000	10,000	96.7	19.76K
Yahoo	100,000	10,000	10,000	79.9	19.73K
SNLI	100,000	10,000	10,000	14.1	9.99K

which can be formalise as

$$\text{sort}(\mathbf{D}^i_{n\times n}) = \text{sort}(\mathbf{D}^l_{n\times n}) = \text{sort}(\mathbf{D}^o_{n\times n}), \tag{2}$$

where $\mathbf{D}^i_{n\times n}$, $\mathbf{D}^l_{n\times n}$ and $\mathbf{D}^o_{n\times n}$ are the adjacency matrices showing observed vector samples' pair-wise distances, with rows and columns aligned. The sort$(\cdot)$ function replaces elements in its input into their row-wise (i.e., sentence-wise in our scenario) rankings the corresponding matrix.

Simple though it is, Eq. (2) can be utilised to evaluate the semantics inconsistency of a VAE's encoder and decoder:

$$\mathbf{M}^E_{n\times n} = \text{sort}(\mathbf{D}^i) - \text{sort}(\mathbf{D}^l), \mathbf{M}^D_{n\times n} = \text{sort}(\mathbf{D}^o) - \text{sort}(\mathbf{D}^l)\,. \tag{3}$$

It is worth noting that in Eq. (3), for each row we only consider the difference corresponding to the k lowest values in $\mathbf{D}^i$, i.e., the k nearest neighbours of the sample investigated for that row. After obtaining $\mathbf{M}^E$ and $\mathbf{M}^D$ which respectively denote the neighbouring structure changes introduced by the encoder and decoder, we then calculate the row-wise sum of $|\mathbf{M}^E|$ and $|\mathbf{M}^D|$, yielding two ranking lists $\mathbf{R}^E$ and $\mathbf{R}^D$), respectively. A row with a larger value in $\mathbf{R}^E$ indicates huger inconsistency between the corresponding input sentence encoding and latent variable's neighbouring structures, which is more likely to correspond to low-density latent regions (and that applies to decoder parallel for $\mathbf{R}^D$). Moreover, the existence of holes can lead to two potential situations which can also be identified using our method. Take $\mathbf{R}^E$ as an example (it is parallel applied to $\mathbf{R}^D$):

1. Large row-wise *negative* values in $\mathbf{R}^E$ (e.g., the i-th row): it means that several *small* values in the i-th row of sort$(\mathbf{D}^i)$ minus the corresponding *large* values in the same row of sort$(\mathbf{D}^l)$, i.e., several semantics-similar sentences regarding $\mathbf{x}_i$ in the input space are mapped to distant regions in the latent space.

2. Large row-wise *positive* values in $\mathbf{R}^E$ (e.g., the i-th row): it means that several *large* values in the i-th row of sort$(\mathbf{D}^i)$ minus the corresponding *small* values in the same row of sort$(\mathbf{D}^l)$, i.e., a local neighbourhood in the latent space are encoding sentences which are originally distant in the input space.

4. Experimental protocol

4.1. Experimental settings

Datasets. We consider three large-scale datasets commonly used for VAE-based language modelling task in previous studies: Yelp 2015 (Yang et al., 2017), Yahoo (Zhang et al., 2015; Yang et al., 2017), and a downsampled version of SNLI (Bowman et al., 2015; Li et al., 2019a). Their statistics is summarised in Tab. 1.

Baselines. To verify the robustness and generalisability of our method, we include five popular architectures for comparison, which are to be pre-trained to converge using hyper-parameters below (they all have official code provided):
- **Basic VAE** (Bowman et al., 2016): using LSTM and KL annealing for mitigating the posterior collapse issue.
- β**-VAE** (Higgins et al., 2017a): utilising an adjustable β to balance the reconstruction loss and the KL term.
- **Cyclical VAE** (Fu et al., 2019): employing cyclical annealing for the KL term.
- **iVAE$_{\text{MI}}$** (Fang et al., 2019): replacing Gaussian-based posteriors with the sample-based distributions.
- **BN-VAE** (Zhu et al., 2020): leveraging the batch normalisation for the variational posterior's parameters.

Hyper-parameters setting. For fair comparison, we follow Kim et al. (2018); He et al. (2019); Fang et al. (2019) to set hyper-parameters. The encoders and decoders of all baselines are constructed using the one-layer LSTM with 1024 hidden dimension and 512-dimensional word embeddings. The dimension of the latent variable is 32. The popular KL annealing strategy (Bowman et al., 2016) is applied, where the scalar weight of the KL term linearly increases from 0 to 1 during the first 10 epochs. Dropout layers with the probability 0.5 are installed on the encoder's both input-to-hidden and hidden-to-output layers. All baselines are trained with Adam optimiser with initial learning rate at 8e-4. The model parameters are initialised using a uniform distribution $U(-0.01, 0.01)$ except word embeddings with $U(-0.1, 0.1)$. The gradients are clipped at 5.0. Early stopping with patience of 5 epochs is adopted when training all models.

4.2. Preliminary experiment: embedding VAE's input and output sentences

Before setting up our method for latent hole detection, we need to first decide the most proper way to form the semantic spaces for input and output sentence encoding (i.e., $\mathbf{X}$ and $\mathbf{X}'$) and calculate similarity matrices. The most straightforward way is to leverage the mean pooling results of the **native word embeddings** (with stop-words excluded) of both trained encoder (for inputs) and decoder (for outputs), respectively. However, very recently Bosc and Vincent (2020) found that even state-of-the-art VAE-based language models tend to memorise the local information (e.g., the first and last words in a sentence) rather than the global one. Based on their observation and insight, we therefore suspect VAE's undesirable memorisation of local information is a potential cause of holes. For the encoder and decoder we hereby consider the embeddings of **the first word**, **the last word**, and **the concatenation of both** as three candidates for feasible encodings of inputs and outputs. Nevertheless, these four listed approaches all ignore important contextualised signals such as bi-grams. Therefore, we also add **BERT embedding** (Xiao, 2018) as the fifth method to consider, which is given by mean pooling over the second last layer of the BERT network (Devlin et al., 2019) and has state-of-the-art performance. To evaluate which encoding strategy to choose, we simply need to see which one leads to the most stable similarity matrices throughout the VAE pipeline (i.e., $|\mathbf{M}^E|$ and $|\mathbf{M}^D|$ in § 3 have small values). Following previous works (vor der Brück and Pouly, 2019; Reimers and Gurevych, 2019), in our experiments we identify vector neighbourhood based on cosine distance.

4.3. Detecting holes and measuring their impact.

After evaluating embedding strategies in § 4.2, we will use the most appropriate one to detect holes in the VAEs' latent spaces and demonstrate how they affect model performance. We will also present results on all other embedding methods for ablation studies.

The core of this experiment lies in the correlation tests on three ranking lists. The first list is the output of our proposed method in § 3, where the samples with higher likelihood of belonging to a hole are assigned with higher ranks. We only consider the top k samples here. Specially, to separately measure latent holes' influence on the encoder and decoder, we will respectively include $\mathbf{R}^E$ and $\mathbf{R}^D$ for comparison. The second list is the automatic evaluation results, where the k samples are ranked based on the perplexity metric. For the third list, we plan to conduct a human evaluation on the sentence quality. More concretely, in each evaluation iteration, we will randomly pick three samples and shuffle them, with the restriction that their rankings have gaps which are at least 10% of k. Next, three human annotators will be invited to rank their quality independently. We will repeat this process for multiple iterations (with duplicated sampling allowed) until enough data is collected. Finally, we will report the correlation coefficients between the first and the second lists, as well the first and the third: the higher they are, to a larger extent the corresponding module (i.e., encoder or decoder) is affected by the existence of holes. Furthermore, if the correlation is consistently strong, we can then recommend our hole-detecting technique to be adopted as a novel quality metric for VAE-based language models.

4.4. Are holes really *vacant*?

Previous studies on image and music have validated the existence of holes in a VAE's latent space, as well as demonstrated that such phenomenon will degrade the models' performance (Rezende and Viola, 2018; Falorsi et al., 2018; Roberts et al., 2018; Kalatzis et al., 2020). They intuitively hypothesised that the variational posterior of low-density latent spaces is close to zero, i.e., no information is learned and the decoded outputs are almost random (Rezende and Viola, 2018; Xu et al., 2019). However, this hypothesis has never been empirically justified. Is it possible that these holes actually capture some signals but in a different (and undesirable) fashion? In this experiment, we will deeply dive into the latent holes and visit this unexplored direction.

We will conduct experiments with two stages. In the first stage, from each of the top k regions which are identified to be of low-density with highest likelihood in § 4.3, we will sample one latent variable (whose coordinate is denoted as c_i (s.t. $i \in [1, k]$)) and decode it into a sentence. Similarly, in the second stage, from the latent space of the *conjugated untrained* VAE model, we will decode the latent variables with coordinates in $\{c_i | i \in [1, k]\}$. For each generated sentence, following (Shannon, 1951) we will calculate its word-level t-gram entropy as

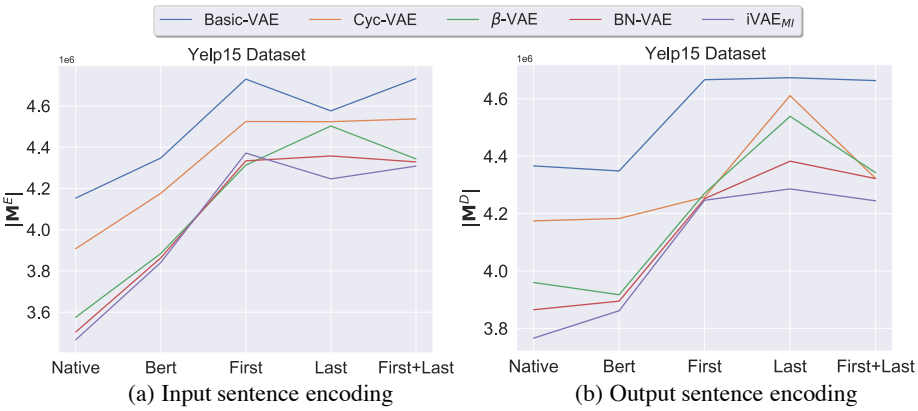
$$\begin{aligned} F_t &= -\sum_{i,j} \text{prob}(b_i, j) \log_2(\text{prob}(b_i, j)/\text{prob}(b_i)) \\ &= -\sum_{i,j} \text{prob}(b_i, j) \log_2 \text{prob}(b_i, j) + \sum_{i} \text{prob}(b_i) \log_2 \text{prob}(b_i), \end{aligned} \tag{4}$$

where b_i is a block of $t-1$ words (i.e., a $(t-1)$-gram), j is an arbitrary word that follows. In such case, prob(b_i) and prob(b_i, j) prob(b_i) respectively denote the probability of b_i and the t-gram $[b_i; j]$. We consider $t \in \{1, 2, 3\}$ in our setup.

For each i, we compare the t-gram entropy of two output sentences in both stages, and use the p-value of two-tailed t-tests with Bonferroni correction (Dror et al., 2018) to examine significance. If the sentences obtained at the first stage have significantly lower entropy than their counterparts at the second, where the decoded latent variables of the both sentences are at the same position (i.e., with same coordinates), then we can show that even the low-density holes actually encode some signals; otherwise they are purely vacant regions and full of randomness.

Table 2: Language modelling results of all pre-trained baselines on the Yelp15, Yahoo and SNLI test datasets. The results of all baselines are reproduced by ourselves. ↓ denotes lower the better and ↑ higher the better.

Model	Yelp15					Yahoo					SNLI				
	Recon↓	PPL↓	MI↑	AU↑	KL	Recon↓	PPL↓	MI↑	AU↑	KL	Recon↓	PPL↓	MI↑	AU↑	KL
Basic VAE	362.61	42.80	2.47	7.00	4.23	331.27	66.19	2.92	8.00	5.95	30.74	23.77	1.98	12.00	3.34
Cyc-VAE	361.13	43.28	3.01	10.00	6.80	331.30	70.42	3.16	15.00	8.49	30.06	24.08	2.46	12.00	4.17
β-VAE (0.4)	353.57	45.12	3.28	17.00	18.43	320.34	74.98	3.28	18.00	24.47	22.53	34.80	3.36	29.00	15.66
BN-VAE (0.7)	351.77	39.94	6.97	32.00	8.32	322.55	63.09	7.16	32.00	8.46	25.85	24.26	7.24	32.00	8.45
iVAE$_{\text{MI}}$	346.31	36.10	3.51	32.00	3.90	305.24	49.41	5.87	32.00	6.26	16.25	8.41	6.54	32.00	6.66

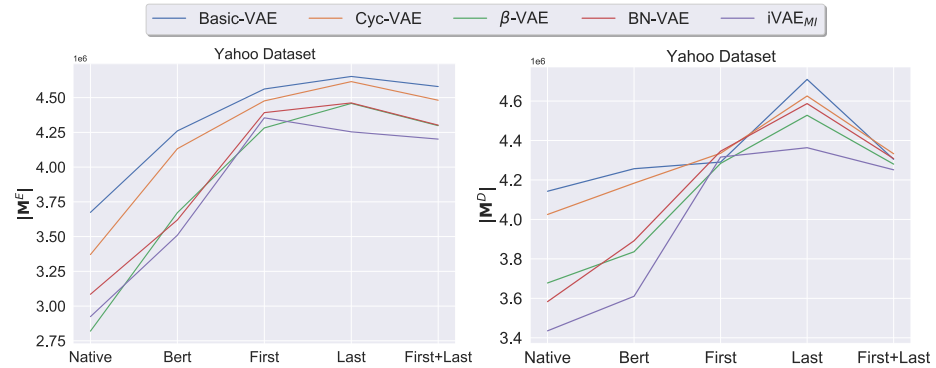


(a) Input sentence encoding (b) Output sentence encoding

Figure 2: The $|\mathbf{M}^E|$ and $|\mathbf{M}^D|$ scores of different embeddings for the input and output sentences on the Yelp15 dataset.

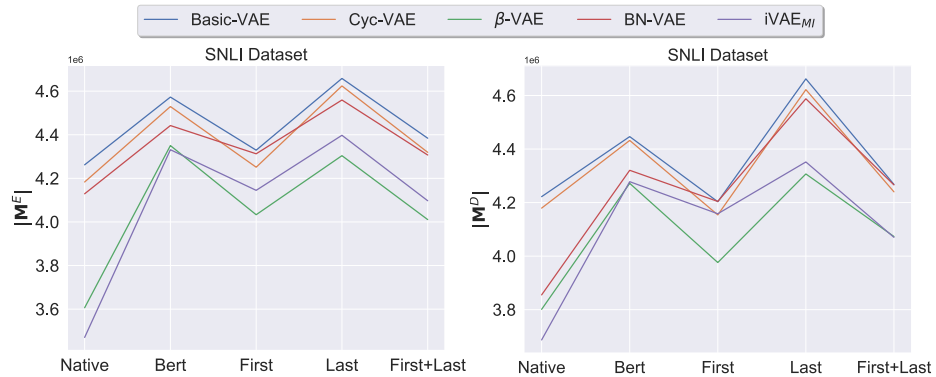
5. Experimental results and analysis

Pre-trained baselines. We train all baselines using the parameters explained in § 4.1 and evaluate baselines using five metrics: reconstruction loss (Recon), perplexity (PPL), mutual information (MI), the number of active units (AU) and KL divergence (KL). Here, MI evaluates how much information of the input $\mathbf{x}$ is captured by the latent variable $\mathbf{z}$; AU



(a) Input sentence encoding (b) Output sentence encoding

Figure 3: The $|\mathbf{M}^E|$ and $|\mathbf{M}^D|$ scores of different embeddings for the input and output sentences in Yahoo dataset.



(a) Input sentence encoding (b) Output sentence encoding

Figure 4: The $|\mathbf{M}^E|$ and $|\mathbf{M}^D|$ scores of different embeddings for the input and output sentences in SNLI dataset.

is the number of active units of the latent variable $\mathbf{z}$, which follows $A_\mathbf{z} = \mathrm{Cov}_\mathbf{x}(\mathbb{E}_{\mathbf{z}\sim q(\mathbf{z}|\mathbf{x})}[z])$ and AU is active if $A_\mathbf{z} > 0.01$.

As shown in Tab. 2, iVAE$_{\mathrm{MI}}$ and Basic VAE achieve the state-of-the-art and worst performance on all datasets, respectively. As for other baselines, the performance of β-VAE and BN-VAE are better than the Cyc-VAE on all datasets regarding the five evaluation metrics. However, β-VAE and BN-VAE have the comparable reconstruction loss on most datasets, although BN-VAE has better performance than β-VAE regarding PPL, MI and AU.

Evaluating different embeddings. As mentioned in § 4.2, we utilise the $|\mathbf{M}^E|$ and $|\mathbf{M}^D|$ in § 3 to evaluate which embedding is a suitable sentence embedding strategy for all baselines. As depicted in Fig. 2, $|\mathbf{M}^E|$ and $|\mathbf{M}^D|$ scores of native and BERT word embeddings are

Table 3: The Spearman's correlation coefficient between the $\mathbf{R}^E/\mathbf{R}^D$ and the perplexity on the encoder (E) and decoder (D) using *native word embedding* for all baselines on three datasets. † and ‡ indicate $p < .5$ and $p < .05$, respectively.

	Yelp15		Yahoo		SNLI	
Model	Native (E)	Native (D)	Native (E)	Native (D)	Native (E)	Native (D)
Basic VAE	$0.025^\dagger$	-0.004	-0.014	$-0.024^\dagger$	-0.004	$0.122^\ddagger$
Cyc-VAE	$0.029^\dagger$	0.015	$0.029^\dagger$	$0.039^\dagger$	0.013	$0.075^\ddagger$
β-VAE	$0.064^\dagger$	0.015	$0.045^\dagger$	$0.027^\dagger$	$0.146^\ddagger$	$0.134^\ddagger$
BN-VAE	$0.011^\dagger$	0.012	0.012	$0.021^\dagger$	$0.105^\ddagger$	$0.139^\ddagger$
iVAE$_{\text{MI}}$	0.005	0.015	$0.070^\ddagger$	0.020	$0.127^\ddagger$	$0.071^\ddagger$

Table 4: The Spearman's correlation coefficient between the $\mathbf{R}^E/\mathbf{R}^D$ and the perplexity on the encoder (E) and decoder (D) using *BERT word embedding* for all baselines on three datasets. † and ‡ indicate $p < .5$ and $p < .05$, respectively.

	Yelp15		Yahoo		SNLI	
Model	Native (E)	Native (D)	Native (E)	Native (D)	Native (E)	Native (D)
Basic VAE	$-0.040^\dagger$	-0.003	0.009	$-0.036^\dagger$	$0.086^\ddagger$	0.014
Cyc-VAE	$-0.026^\dagger$	$0.021^\dagger$	$-0.031^\dagger$	$0.039^\dagger$	$0.024^\dagger$	$-0.032^\dagger$
β-VAE	-0.014	$0.124^\ddagger$	$-0.026^\dagger$	0.013	$0.088^\ddagger$	$0.081^\ddagger$
BN-VAE	$0.014^\dagger$	$0.022^\dagger$	-0.031	$-0.011^\dagger$	$0.065^\ddagger$	$0.089^\ddagger$
iVAE$_{\text{MI}}$	$0.037^\dagger$	$0.033^\dagger$	$-0.043^\dagger$	$-0.022^\dagger$	$0.063^\ddagger$	$0.104^\ddagger$

obviously lower than the scores of the first word, last word and the concatenation of both on all baselines, which shows that the first word, the last word and both of them are not the ideal sentence embedding strategy (other datasets have the similar trend in Fig. 3 and Fig. 4). After analysing the first and last words occurring in the Yelp15 test dataset, we find that most words are the *article*, *adjective* and common *noun* (e.g., i, this, good, glad, food, etc), which cannot accurately express the meaning of the entire sentence (other datasets have the similar issue). In contrast with the BERT embedding, the native word embedding achieves the better performance for all baselines. The reason why BERT embedding is inferior might be that the position of the mean pooling BERT embedding for the input and output sentences in the embedding space are disturbed by the extremely external large-scale corpus used in the pre-trained BERT embedding. However, since the native word embedding is trained with the VAE from scratch, the position of the mean pooling sentence embedding is not seriously affected by the external resources.

Detecting holes. After selecting the native word embedding as our sentence embedding strategy, we respectively conduct the Spearman's correlation tests on three ranking lists (cf. § 4.3). We consider the top $k = 1000$ samples from $\mathbf{R}^E$ and $\mathbf{R}^D$ here. As shown in Tab. 3, the Spearman's correlation coefficient between the $\mathbf{R}^E/\mathbf{R}^D$ and perplexity of the samples for most baselines are weakly positive (with only few $p < .05$), which means that a

Table 5: The Spearman's correlation coefficient between the $\mathbf{R}^E/\mathbf{R}^D$ and the perplexity on the encoder (E) and decoder (D) using *first word embedding* for all baselines on three datasets. † and ‡ indicate $p < .5$ and $p < .05$, respectively.

	Yelp15		Yahoo		SNLI	
Model	Native (E)	Native (D)	Native (E)	Native (D)	Native (E)	Native (D)
Basic VAE	0.028†	-0.073†	0.074‡	-0.097‡	0.066‡	0.032†
Cyc-VAE	0.006	0.028†	0.051†	0.109‡	-0.005	-0.003
β-VAE	-0.014	0.033†	0.071‡	0.053†	0.145‡	0.130‡
BN-VAE	0.013	0.085†	0.046	0.048‡	0.067‡	0.069‡
iVAE$_{\text{MI}}$	0.022†	0.162‡	-0.001	-0.019	0.077‡	0.072‡

Table 6: The Spearman's correlation coefficient between the $\mathbf{R}^E/\mathbf{R}^D$ and the perplexity on the encoder (E) and decoder (D) using *last word embedding* for all baselines on three datasets. † and ‡ indicate $p < .5$ and $p < .05$, respectively.

	Yelp15		Yahoo		SNLI	
Model	Native (E)	Native (D)	Native (E)	Native (D)	Native (E)	Native (D)
Basic VAE	-0.035†	0.037†	0.008	-0.052†	0.030†	0.040†
Cyc-VAE	0.015	0.044†	0.035†	0.081‡	0.070‡	0.012
β-VAE	-0.048†	-0.036†	-0.021	0.001	0.042†	0.121‡
BN-VAE	-0.057†	0.075†	0.011	0.018†	0.065‡	0.134‡
iVAE$_{\text{MI}}$	-0.067‡	0.083‡	0.012	0.024†	0.079‡	0.178‡

higher score in $\mathbf{R}^E$ or $\mathbf{R}^D$ (i.e., a higher likelihood for a latent hole) might not has a larger perplexity for the corresponding sentence (i.e., a worse quality for the sentence). We further conduct the human evaluation on the 1000 samples from $\mathbf{R}^E$ and $\mathbf{R}^D$, where the Basic VAE, β-VAE and iVAE$_{\text{MI}}$ are evaluated on Yelp15 and SLNI datasets (cf. § 4.3). Following the setting in § 4.3, we iteratively repeat the random sampling 30 times (i.e., 30 groups where each of them contains 3 shuffled samples) in the $\mathbf{R}^E/\mathbf{R}^D$ for each model and dataset. Then three postgraduate students are invited to rank the quality of the three shuffled samples in each group independently. The Light's Kappa (Light, 1971; Conger, 1980) is used to evaluate the inter-rater agreement between three annotators and the average Kappa is only 0.289, which means that the agreement among the annotators is fairly weak and thus implies that there might be no strong correlation. As shown in Tab. 8, although there exists moderate positive correlation on the encoder side for all baselines on the Yelp15 and SNLI datasets, the Kappa score is lower than 0.3 and therefore the positive correlation result is not statistically significant. For the decoder side, the human evaluation result agrees with the results in Tab. 3, i.e., there is no obvious positive or negative correlation between the large latent hole and the quality of the sentences. We further conduct the ablation studies of the Spearman's correlation test for all other embedding methods in Tab. 4, 5, 6 and 7 which also show the same trend, i.e., the strong positive or negative correlation is not statistically significant.

Table 7: The Spearman's correlation coefficient between the $\mathbf{R}^E/\mathbf{R}^D$ and the perplexity on the encoder (E) and decoder (D) using *first+last word embedding* for all baselines on three datasets. † and ‡ indicate $p < .5$ and $p < .05$, respectively.

	Yelp15		Yahoo		SNLI	
Model	Native (E)	Native (D)	Native (E)	Native (D)	Native (E)	Native (D)
Basic VAE	0.003	-0.012	0.006	-0.034†	0.033†	-0.038†
Cyc-VAE	-0.006	0.050†	0.022†	0.137‡	-0.032†	-0.027†
β-VAE	-0.035†	-0.054†	0.036†	0.015	0.163‡	-0.005
BN-VAE	0.048†	0.001	0.029	0.002	0.087‡	0.102‡
iVAE$_{\text{MI}}$	0.062‡	0.008	-0.019	-0.002	0.115‡	0.129‡

Table 8: The Spearman's correlation coefficient between human ranking list and the corresponding $\mathbf{R}^E/\mathbf{R}^D$ on the encoder (E) and decoder (D) using *native word embedding* for three on Yelp15 and SNLI datasets. † and ‡ indicate $p < .5$ and $p < .05$, respectively.

	Yelp15		SNLI	
Model	Native (E)	Native (D)	Native (E)	Native (D)
Basic VAE	0.300‡	0.078†	0.561‡	0.061
β-VAE	0.383‡	0.011	0.402‡	-0.172†
iVAE$_{\text{MI}}$	0.356‡	-0.106†	0.127‡	-0.300‡

Are holes really vacant? Based on our hypothesis in § 3, there will be some large areas in the latent space in which semantics-distant sentences can be mapped. Consequently, those areas yield large values in $\mathbf{R}^E$ and $\mathbf{R}^D$ correspond to generated sentences with high perplexity, i.e., strong positive correlation between the $\mathbf{R}^E/\mathbf{R}^D$ and the perplexity. However, the negative experiment results shows that our large latent holes hypothesis might not be valid. That is, there is no strong correlation between the size of latent holes and the sentence quality. Therefore, the experiment to explore whether the holes are really vacant or not based on our hypothesis cannot be concluded. Nevertheless, a different hypothesis can be derived from the invalid large latent hole hypothesis based on our experiments above. That is, latent holes might be densely distributed in the latent space and the size of each hole is tiny. In the future, we plan to design different experiments to explore the small and ubiquitous latent hole hypothesis.

6. Conclusion

In this paper, we are the first work to explore how the latent holes distribute and how they affect the VAE by proposing a simple hole-detection algorithm based on the neighbour consistency between VAE's input, latent and output semantic space. The large-scale low-density latent holes might not exist in the latent space after comprehensive experiments. We also conduct experiments to evaluate which sentence embedding strategy suits the VAE,

and the experiments show that native word embedding is an ideal embedding strategy for the VAE in the language modelling task. Since the large latent hole hypothesis might not be valid based on our experiments, we plan to explore whether the latent holes are densely distributed in the latent space and propose solutions to improve the VAE training in the future. Another direction we are interested to explore is to test our frame work in the security domain, such as improving the classification effectiveness of intrusion detection for VAE based methods.

Acknowledgement

This work is supported by the award made by the UK Engineering and Physical Sciences Research Council (Grant number: EP/P011829/1) and Ningbo Natural Science Foundation (202003N4320, 202003N4321). We would like to thank all the anonymous reviewers for their insightful and helpful comments.

References

Tom Bosc and Pascal Vincent. Do sequence-to-sequence VAEs learn global features of sentences? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4296–4318, Online, November 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-main.350.

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. In *Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*, pages 632–642. Association for Computational Linguistics (ACL), 2015.

Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, 2016.

Anthony J Conger. Integration and generalization of kappas for multiple raters. *Psychological Bulletin*, 88(2):322, 1980.

Tim R. Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M. Tomczak. Hyperspherical variational auto-encoders. *34th Conference on Uncertainty in Artificial Intelligence (UAI-18)*, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th*

Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1383–1392. Association for Computational Linguistics, 2018.

Luca Falorsi, Pim de Haan, Tim R Davidson, Nicola De Cao, Maurice Weiler, Patrick Forré, and Taco S Cohen. Explorations in homeomorphic variational auto-encoding. *arXiv preprint arXiv:1807.04689*, 2018.

Le Fang, Chunyuan Li, Jianfeng Gao, Wen Dong, and Changyou Chen. Implicit deep latent variable models for text generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3937–3947, 2019.

Hao Fu, Chunyuan Li, Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, and Lawrence Carin. Cyclical annealing schedule: A simple approach to mitigating kl vanishing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 240–250, 2019.

Junxian He, Daniel Spokoyny, Graham Neubig, and Taylor Berg-Kirkpatrick. Lagging inference networks and posterior collapse in variational autoencoders. In *International Conference on Learning Representations*, 2019.

I. Higgins, Loïc Matthey, A. Pal, C. Burgess, Xavier Glorot, M. Botvinick, S. Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017a.

Irina Higgins, Arka Pal, Andrei Rusu, Loic Matthey, Christopher Burgess, Alexander Pritzel, Matthew Botvinick, Charles Blundell, and Alexander Lerchner. Darla: Improving zero-shot transfer in reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1480–1490. JMLR.org, 2017b.

Huaibo Huang, Ran He, Zhenan Sun, Tieniu Tan, et al. Introvae: Introspective variational autoencoders for photographic image synthesis. In *Advances in Neural Information Processing Systems*, pages 52–63, 2018.

Dimitris Kalatzis, David Eklund, Georgios Arvanitidis, and Søren Hauberg. Variational autoencoders with riemannian brownian motion priors. *arXiv preprint arXiv:2002.05227*, 2020.

Yoon Kim, Sam Wiseman, Andrew Miller, David Sontag, and Alexander Rush. Semi-amortized variational autoencoders. In *International Conference on Machine Learning*, pages 2678–2687, 2018.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Bohan Li, Junxian He, Graham Neubig, Taylor Berg-Kirkpatrick, and Yiming Yang. A surprisingly effective fix for deep latent variable modeling of text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th*

International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3594–3605, 2019a.

Chunyuan Li, Xiang Gao, Yuan Li, Xiujun Li, Baolin Peng, Yizhe Zhang, and Jianfeng Gao. Optimus: Organizing sentences via pre-trained modeling of a latent space. *arXiv preprint arXiv:2004.04092*, 2020a.

Ruizhe Li, Xiao Li, Chenghua Lin, Matthew Collinson, and Rui Mao. A stable variational autoencoder for text modelling. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 594–599, 2019b.

Ruizhe Li, Xiao Li, Guanyi Chen, and Chenghua Lin. Improving variational autoencoder for text modelling with timestep-wise regularisation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2381–2397, 2020b.

Xiao Li, Chenghua Lin, Ruizhe Li, Chaozheng Wang, and Frank Guerin. Latent space factorisation and manipulation via matrix subspace projection. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5916–5926. PMLR, 13–18 Jul 2020c. URL http://proceedings.mlr.press/v119/li20i.html.

Richard J Light. Measures of response agreement for qualitative data: some generalizations and alternatives. *Psychological bulletin*, 76(5):365, 1971.

Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *Advances in Neural Information Processing Systems*, pages 14837–14847, 2019.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3973–3983, 2019.

Danilo Jimenez Rezende and Fabio Viola. Taming vaes. *arXiv preprint arXiv:1810.00597*, 2018.

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

Adam Roberts, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. A hierarchical latent vector model for learning long-term structure in music. In *International Conference on Machine Learning*, pages 4364–4373, 2018.

Claude E Shannon. Prediction and entropy of printed english. *Bell system technical journal*, 30(1):50–64, 1951.

Dinghan Shen, Asli Celikyilmaz, Yizhe Zhang, Liqun Chen, Xin Wang, Jianfeng Gao, and Lawrence Carin. Towards generating long and coherent text with multi-level latent variable

models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2079–2089, 2019.

Tim vor der Brück and Marc Pouly. Text similarity estimation based on word embeddings and matrix norms for targeted marketing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1827–1836, 2019.

Han Xiao. bert-as-service. https://github.com/hanxiao/bert-as-service, 2018.

Peng Xu, Jackie Chi Kit Cheung, and Yanshuai Cao. On variational learning of controllable representations for text without supervision. *arXiv preprint arXiv:1905.11975*, 2019.

Weidi Xu, Haoze Sun, Chao Deng, and Ying Tan. Variational autoencoder for semi-supervised text classification. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 3358–3364, 2017.

Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. Improved variational autoencoders for text modeling using dilated convolutions. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3881–3890. JMLR. org, 2017.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015.

Tiancheng Zhao, Kyusong Lee, and Maxine Eskenazi. Unsupervised discrete sentence representation learning for interpretable neural dialog generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1098–1107, 2018.

Qile Zhu, Wei Bi, Xiaojiang Liu, Xiyao Ma, Xiaolin Li, and Dapeng Wu. A batch normalized inference network keeps the KL vanishing away. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2636–2649, Online, July 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.acl-main.235.