On Variational Learning of Controllable Representations for Text without Supervision

Peng Xu [1] **Jackie Chi Kit Cheung** [1 2 3] **Yanshuai Cao** [1]

Abstract

The variational autoencoder (VAE) can learn the manifold of natural images on certain datasets, as evidenced by meaningful interpolation or extrapolation in the continuous latent space. However, on discrete data such as text, it is unclear if unsupervised learning can discover a similar latent space that allows controllable manipulation. In this work, we find that sequence VAEs trained on text fail to properly decode when the latent codes are manipulated, because the modified codes often land in holes or vacant regions in the aggregated posterior latent space, where the decoding network fails to generalize. Both as a validation of the explanation and as a fix to the problem, we propose to constrain the posterior mean to a learned probability simplex, and perform manipulation within this simplex. Our proposed method mitigates the latent vacancy problem and achieves the first success in unsupervised learning of controllable representations for text. Empirically, our method outperforms unsupervised baselines and strong supervised approaches on text style transfer, and is capable of performing more flexible fine-grained control over text generation than existing methods.

1. Introduction

High-dimensional data, such as images and text, are often generated through the interaction of many complex factors, such as lighting and pose in images or style and content in texts. Recently, VAEs and other unsupervised generative models have found successes in modelling the manifold of natural images (Higgins et al., 2017; Kumar et al., 2017; Chen et al., 2016). These models often discover controllable latent factors that allow manipulation of the images through conditional generation from interpolated or extrapolated latent codes, often with impressive quality. On the other hand, while various attributes of text such as sentiment and topic can be discovered in an unsupervised way, manipulating the text by changing these learned factors has not been possible with unsupervised generative models, to the best of our knowledge. Cífka et al. (2018); Zhao et al. (2018) observed that text manipulation is generally more challenging compared to images, and the successes of these models cannot be directly transferred to texts.

Controllable text generation aims at generating realistic text with control over various attributes including sentiment, topic and other high-level properties. The possibility of unsupervised controllable text generation could help in a wide range of applications such as dialogues systems (Wen et al., 2016). Existing approaches (Shen et al., 2017; Fu et al., 2018; Li et al., 2018; Sudhakar et al., 2019) all rely on supervised learning from annotated attributes to generate the text in a controllable fashion. The high cost of labelling large training corpora with attributes of interest limits the usage of these models, as pre-existing annotations often do not align with desired downstream goals. Even if cheap labels are available, for example, review scores as a proxy for sentiment, the control is limited to the variation defined by the attributes.

In this work, we examine the obstacles that prevent sequence VAEs (Bowman et al., 2015) from performing well in unsupervised controllable text generation. We empirically discover that manipulating the latent factors for typical semantic variations often leads to latent codes that reside in some low-density region of the aggregated posterior distribution. In other words, there are *vacant* regions in the latent code space (Makhzani et al., 2015; Rezende & Viola, 2018) not being considered by the decoding network, at least not at convergence. As a result, the decoding network is unable to process such manipulated latent codes, yielding unpredictable generation results of low quality. Although this issue has been raised in prior works, we provide direct evidence using topological data analysis to show that this vacancy problem is more severe for VAEs trained on text than image.

In order to mitigate the latent vacancy problem on text, we

[1]Borealis AI [2]McGill University [3]Canada CIFAR Chair, Mila. Correspondence to: Peng Xu <peng.z.xu@borealisai.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

propose to constrain the posterior mean to a learned probability simplex and only perform manipulation within the probability simplex, which is referred as CP-VAE (Constrained Posterior VAE). Two regularizers are added to the original objective of VAE. The first enforces an orthogonal structure of the learned probability simplex; the other encourages this simplex to be filled without holes. Besides confirming that latent vacancy is indeed a cause of failure in previous sequence VAEs', CP-VAE is also the first successful attempt towards unsupervised learning of controllable representations for text to the best of our knowledge. Experimental results on text style transfer show that our method outperforms unsupervised baselines and strong supervised approaches, whose decoding network are trained from scratch. Without supervision and the help of pre-training for generation, our method achieves comparable results with state-of-the-art supervised approaches leveraging large-scale pre-trained models for generation, with respect to the automatic evaluation metrics used in text style transfer. Our proposed framework also enables finer-grained and more flexible control over text generation. In particular, we can switch the topic in the middle of sentence generation, and the model will often still find a way to complete the sentence in a natural way, which has never been attempted by previous methods.[1]

2. Background: Variational Autoencoders

The variational autoencoder (VAE) (Kingma & Welling, 2013) is a generative model defined by a prior $p(\boldsymbol{z})$ and a conditional distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$. The VAE is trained to optimize a tractable variational lower bound of $\log p_{\boldsymbol{\theta}}(\boldsymbol{x})$:

$$\begin{aligned}\mathcal{L}_{\text{VAE}}(\boldsymbol{x};\boldsymbol{\theta},\boldsymbol{\phi}) =& \mathbf{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})] \\ &- D_{\text{KL}}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})),\end{aligned} \tag{1}$$

where $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ is a variational distribution parameterized by an encoding network with parameters $\boldsymbol{\phi}$, and $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ denotes the decoding network with parameters $\boldsymbol{\theta}$. This objective tries to minimize the reconstruction error to generate the data, and at the same time regularizes $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ towards the prior $p(\boldsymbol{z})$. For text modelling, the input $\boldsymbol{x}$ is some observed text. Both the encoding and decoding network are usually recurrent neural networks, and the model is called a sequence VAE.

Note that during learning, the decoding network $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ only learns to decode $\boldsymbol{z}$ that are sampled from $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$. *In other words, the decoding network is never trained to decode the entire latent space.* Instead, it only learns to process $\boldsymbol{z}$ sampled from the aggregated posterior distribution $q_{\boldsymbol{\phi}}(\boldsymbol{z}) = \mathbf{E}_{\boldsymbol{x}\sim p_d(\boldsymbol{x})} q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$, where $p_d(\boldsymbol{x})$ is the training data distribution. If $q_{\boldsymbol{\phi}}(\boldsymbol{z})$ has regions of low density, there is no guarantee that $p_{\boldsymbol{\theta}}$ would generalize well to such places. This is an important intuition that will become central to our analysis in Sec. 3.

[1]The code to reproduce our results can be found in https://github.com/BorealisAI/CP-VAE

3. Latent Vacancy Hypothesis

We hypothesize that when trained on text data, the aggregated posterior of sequence-VAEs tend to have vacant regions of low density, where the decoder may fail to generalize to. The decoder could generalize to the vacant regions without ever seeing training examples, but there is no guarantee it can perform well in this case especially if the such vacancy is large. Fig. 1 is an illustration of the intuition.

In this section, we conduct exploratory study on unsupervised sentiment manipulation and provide evidence from two different aspects to verify the above-mentioned hypothesis. First, we measure how the negative log-likelihood of latent codes under the aggregated posterior changes before and after manipulation. Second, since topology is the technical language to describe the notion of vacant regions or holes, we employ topological data analysis to confirm the exacerbation of latent vacancy problem on text as compared to images. In addition, we give a preview of our proposed method (later formally introduced in Section 4) and demonstrate that it avoids the latent vacancy problem using the same analyses.

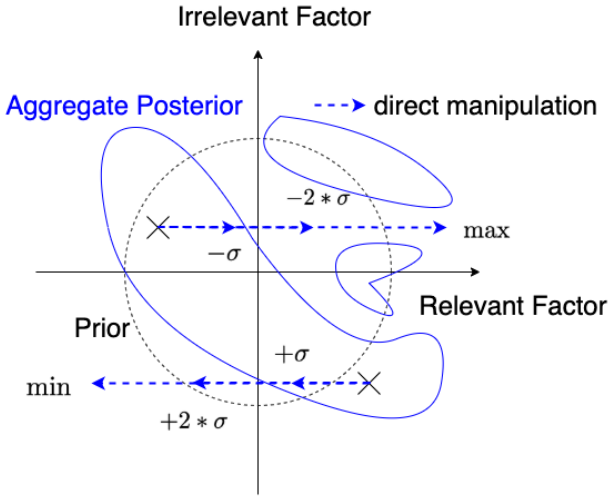

Figure 1. Illustration of why latent vacancy prevents effective manipulation in VAEs. The aggregated posterior shown has multiple disconnected areas and direct manipulations of the relevant factor may fall into vacant regions of low density.

3.1. Unsupervised Sentiment Manipulation

Here we describe the setup used to discover a sentiment latent dimension and subsequent exploration of manipulating the sentiment attribute. Note that discovering sentiment

	Example	Transfer Strength	Content Preservation	NLL Discrepancy
Source sentence	the pizza is offered without toppings and it 's lacking in flavor .	-	-	-
β-VAE w. aggr training ($\pm\sigma$)	the pizza is offered in toppings and it 's lacking in pittsburgh sauce .	Weak	Good	Small
β-VAE w. aggr training ($\pm 2 * \sigma$)	the pizza is more than fresh and your food is lacking in flavor	Medium	Medium	Medium
β-VAE w. aggr training (extremum)	the service is a great cut and the food is top notch in charlotte .	Strong	Bad	Large
CP-VAE (this work)	the pizza is full of spicy and it 's delicious .	Strong	Good	Small

Table 1. Summary of the behaviours of β-VAE with aggressive training and our proposed CP-VAE. Detailed quantitative evaluations for transfer strength and content preservation are presented in Tab. 2.

feature in an unsupervised way is known to be possible, *e.g.*, in large-scale language models (Radford et al., 2017). However, limited success has been achieved for sequence VAE and its variants to change text attributes while preserving the relevant content, without annotated labels.

To perform unsupervised sentiment manipulation, we use the Yelp restaurant reviews dataset and the same data split following Li et al. (2018). We train a β-VAE (Higgins et al., 2017) with aggressive training of the encoder as proposed by He et al. (2019), which is the state of the art, and a significant improvement over vanilla sequence VAEs. The model under study here has a latent space of 80 dimensions with a LSTM encoder and decoder, with a β of 0.35. By inspecting the accuracy on the validation set, we find that there exists one dimension of latent code, $\boldsymbol{z}_{[s]}$, achieving around 75% sentiment classification accuracy by its value alone, while other latent codes get accuracy around 50%. This means that this latent dimension is an effective sentiment indicator. Further details can be found in Appendix B.1-B.2.

However, when we try to perform sentiment manipulation by modifying this latent dimension, the decoding network fails to generate desirable outputs most of the time. To ensure that the magnitude of manipulation suffices to change the sentiment of generated text, we try multiple magnitudes by moving $\boldsymbol{z}_{[s]}$ (1) by σ; (2) by $2 * \sigma$; (3) to $\min(\boldsymbol{z}_{[s]})$ or $\max(\boldsymbol{z}_{[s]})$, where σ, $\min$, $\max$ are the the standard deviation, the minimum and the maximum estimated on all the training samples. How we conduct the manipulation is illustrated in Fig. 1. We inspect the generated sentences with the manipulated codes to check whether they are transferred to the desired style successfully (*transfer strength*) and whether they are still relevant to the source sentence (*content preservation*). We summarize the behaviours of β-VAE with aggressive training in Tab. 1, along with one randomly selected example for the purpose of illustration. Although the sentiment can be flipped as we increase the magnitude of the manipulations, the transformed texts become irrelevant to the original text, meaning the content information in the latent code is ignored by the decoder. On the other hand, when the manipulation on $\boldsymbol{z}_{[s]}$ is small as in Fig. 2 (A), β-VAE is unable to flip the sentiment of the transformed text, like the example in Tab. 1. Detailed quantitative evaluations are presented in Sec. 5.1.

3.2. NLL of the Codes under the Aggregated Posterior

To verify our hypothesis of vacant regions, we first compare the negative log-likelihood (NLL) of test samples' original latent codes as well as the manipulated ones, under the aggregated posterior. An increase of the NLL after manipulation would indicate that the new codes land in regions of lower density. The aggregated posterior of our trained VAE is estimated with a large mixture of Gaussians where each component is the Gaussian posterior at one training data point. Each test point's code (taken posterior mean) has an NLL under this mixture density. Fig. 2 shows the histograms of NLLs of all 1000 test samples' codes before and after manipulation. We can see that the discrepancy in NLL between the original and the manipulated codes becomes larger as we increase the magnitude of the manipulation, indicating that the manipulated codes may fall into the low density area.

3.3. Highest Density Region and Topological Analysis

The notion of vacant regions or holes is a topological concept, so it is natural to use tools from topological data analysis (TDA) to measure and visualize this phenomenon. Given the aggregated posterior $q_{\boldsymbol{\phi}}(\boldsymbol{z})$, the highest density region (HDR) at level $(1-\epsilon)$ (Hyndman, 1996) is defined to be: $D_\epsilon = \{\boldsymbol{z} | q_{\boldsymbol{\phi}}(\boldsymbol{z}) \geq c_\epsilon\}$, where c_ϵ is the largest constant such that $Pr(z \in D_\epsilon) \geq 1-\epsilon$. Intuitively HDR captures the notion of "significant support", where we cut the density at c_ϵ to form a subset D_ϵ of the latent space that contains at least $1-\epsilon$ of the probability mass. What we mean by the vacancy in the aggregated posterior $q_{\boldsymbol{\phi}}(\boldsymbol{z})$ is that the $(1-\epsilon)$-HDR has holes or disconnected components. We want to emphasize that ϵ is conceptual and used to formalize the definition; it is not a hyperparameter of any model. In practice, whenever we draw a finite sample set, the points

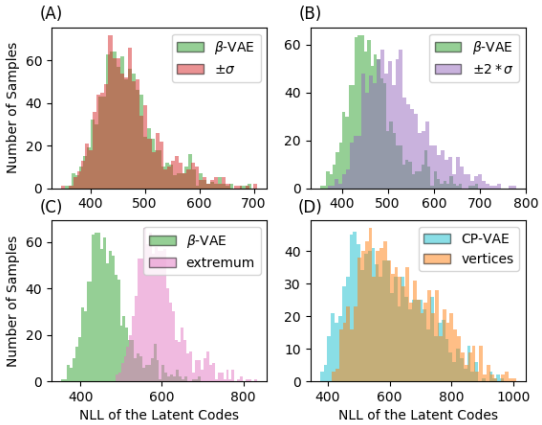

Figure 2. Histograms of all the test samples' negative log-likelihood (NLL) under the aggregated posterior, considering their original latent codes and manipulated ones. (A) (B) (C): three manipulation strategies for β-VAE with aggressive training; (D) CP-VAE.

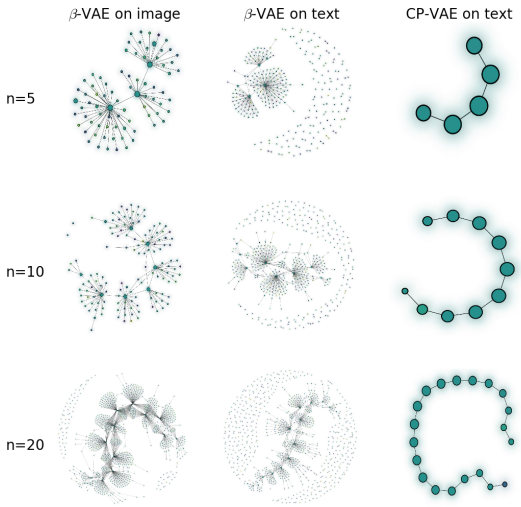

Figure 3. Topological analysis of the highest density region (HDR) of aggregated posterior using the mapper algorithm. The connectedness of the graph holds the key topological information; the shape on the 2D plane is irrelevant. Different n's control the coarseness of visualization. If a structure persists at multiple resolutions, it is stable. If it appears and disappears for selected value or a small range of n, then it is likely to be "topological noise".

are in the HDR D_ϵ with probability $1-\epsilon$, for some strictly positive ϵ.

We use the mapper algorithm (Singh et al., 2007) here to visualize the connectedness of D_ϵ's[2] for β-VAE trained on images and text respectively. Further details can be found in Appendix B. The input to the mapper algorithm is a point cloud. For us, it is the posterior samples at training points under each model. The output of the mapper is a graph, like the ones in Figure 3. Each node in the graph corresponds to a set of nearby points in the original point cloud. The connectivity of the graph reflects some topological properties of the sampling space of the point cloud. Such properties include connectedness and the presence of holes.

The main take-away, as shown in Fig. 3, is that the HDR of β-VAE on images is one connected component (up to topological noise on the finest scale); whereas, for text, there are many disconnected components across all scales of visualization. This observation suggests that the underlying D_ϵ for β-VAE on text is disconnected, providing empirical evidence that the latent vacancy problem is more severe on text than on images. Further explanations about the relationship of connectedness of D_ϵ and that of the mapper graphs can be found in Appendix B.4.

3.4. Constraining the posterior

In order to resolve the latent vacancy problem, we propose CP-VAE in this work, where we constrain the posterior in such a way that the manipulation only happens in a learned simplex, so that most space in this constrained subspace can be covered during training. In this constrained subspace, the phenomenon of low density holes of aggregated posterior is significantly reduced, as Fig. 2 (D) empirically show that there is little change in NLL of original versus manipulated codes. Furthermore, Fig. 3 shows that the HDR of CP-CAE is one connected component[3]. At the same time, CP-VAE can maintain its transfer strength to effectively transfer the source sentence to the desired style, as exemplified in Tab. 1. The details of our method are presented in the next section.

4. Method

4.1. Overview

The experiments conducted in Sec. 3 validate the existence of vacancy in the aggregated posterior latent space. One potential way to resolve the problem is to better match the aggregated posterior with the prior (Makhzani et al., 2015; Tomczak & Welling, 2017; Zhao et al., 2018). However, in terms of unsupervised learning of controllable representation for text, these previous methods have not shown success; Zhao et al. (2018) only attempted supervised text style transfer, and also reported negative results from the AAE

[2]In practice, we use the Kepler Mapper library by van Veen et al. (2019)

[3]The HDR visualized here is for $\boldsymbol{z}^{(1)}$ introduced in Sec. 4

(Makhzani et al., 2015). Another way to resolve the vacancy issue is to directly enforce that the aggregated posterior itself has no vacant region anywhere where we would like to perform latent code manipulation. We propose to map the posterior Gaussian mean to a constrained space, more specifically a learned probability simplex, where we can encourage the constrained latent space to be filled without vacancy, and perform manipulation to be within this simplex. We add a mapping function as part of the encoding network which maps the mean of the Gaussian posterior to a constrained space. Two regularization terms are introduced to ensure the learned simplex is not degenerate and that this subspace is well filled.

In addition, we model the relevant factors that we wish to control separated from the irrelevant factors by splitting $\boldsymbol{z}$ into two parts, $\boldsymbol{z}^{(1)}$ and $\boldsymbol{z}^{(2)}$, following prior work (Bao et al., 2019). The first part captures the relevant factors that are dominant in the data without an inductive bias from external signals, while the second part learns to encode the remaining local information that is useful for reconstructing the source sentences. As a result, $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ is decomposed into $q_{\boldsymbol{\phi}_1}(\boldsymbol{z}^{(1)}|\boldsymbol{x})q_{\boldsymbol{\phi}_2}(\boldsymbol{z}^{(2)}|\boldsymbol{x})$ where $\boldsymbol{\phi} = \boldsymbol{\phi}_1 \cup \boldsymbol{\phi}_2$. With diagonal covariances, the KL divergence term in Eq. 1 splits into two separate KL terms. In practice, we use a MLP encoding network to parametrize $\boldsymbol{z}^{(1)}$ with some sentence representation as the input (*e.g.*, averaging GloVe embeddings (Pennington et al., 2014) over the input tokens) and a LSTM encoding network to parametrize $\boldsymbol{z}^{(2)}$. We only constrain the posterior of $\boldsymbol{z}^{(1)}$, and $\boldsymbol{z}^{(2)}$ is optimized the same way as the traditional VAE.

4.2. Constraining the Posterior

We now describe how to map the mean $\boldsymbol{\mu}$ of the Gaussian posterior for $\boldsymbol{z}^{(1)} \in \mathbb{R}^N$ to a constrained latent space. We would like to constrain the mean $\boldsymbol{\mu}$ to have a structure as follows:

$$\boldsymbol{\mu} = \sum_{i=1}^{K} p_i \boldsymbol{e}_i, \sum_{i=1}^{K} p_i = 1, \langle \boldsymbol{e}_i, \boldsymbol{e}_j \rangle = 0, i \neq j, K \leq N \quad (2)$$

where $\boldsymbol{e}_i$ are vectors representing the relevant factors, p_i is the proportion of ith relevant factor encoded in $\boldsymbol{z}^{(1)}$ and K is a hyperparameter indicating the number of relevant factors to discover. In other words, the mean of the Gaussian posterior of $\boldsymbol{z}^{(1)}$ is constrained to be inside a K-dimension probability simplex in $\mathbb{R}^N$ whose vertices are represented by the orthogonal basis vectors $\boldsymbol{e}_i, i = 1, \ldots, K$. Given the outputs of the MLP encoder $\boldsymbol{h}$ and $\log \boldsymbol{\sigma}^2$, we learn an additional mapping function π which maps $\boldsymbol{h}$ to the constrained posterior space, which can be treated as part of the encoding network:

$$\boldsymbol{\mu} = \pi(\boldsymbol{h}) = \boldsymbol{E} \cdot \text{softmax}(\boldsymbol{W}\boldsymbol{h} + \boldsymbol{b}), \quad (3)$$

where $\boldsymbol{E} = [\boldsymbol{e}_1, \ldots, \boldsymbol{e}_K]$ is a learnable embedding matrix representing the bases, $\boldsymbol{W}$ is the learnable weight matrix, and $\boldsymbol{b}$ is the learnable bias vector. As a result, the constrained posterior is parametrized by $\boldsymbol{\mu}$ and $\log \boldsymbol{\sigma}^2$ as a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma}^2))$.

With the mapping function alone, the proposed VAE suffers from posterior collapse (Bowman et al., 2015), a well-known problem where the model ignores the latent code $\boldsymbol{z}$ during the training. Further complicating matters is the fact that there is an abundance of signals for predicting the next token in the text, but the signals indicating high-level semantics are quite sparse. It is thus unlikely that the VAEs can capture useful relevant factors from raw text without collapse. For these reasons, we enforce orthogonality in the learnt basis vectors as defined in Eq. 2, which introduces a natural recipe to prevent posterior collapse for $\boldsymbol{z}^{(1)}$. Note that the KL divergence between $q_{\boldsymbol{\phi}_1}(\boldsymbol{z}^{(1)}|\boldsymbol{x})$ and $p(\boldsymbol{z}^{(1)})$ is

$$\begin{aligned} &D_{\text{KL}}(q_{\boldsymbol{\phi}_1}(\boldsymbol{z}^{(1)}|\boldsymbol{x}) \| p(\boldsymbol{z}^{(1)})) = \\ &\quad \frac{1}{2}\boldsymbol{\mu}^\top\boldsymbol{\mu} + \frac{1}{2}\left(\boldsymbol{\sigma}^\top\boldsymbol{\sigma} - \log \boldsymbol{\sigma}^\top\boldsymbol{\sigma} - 1\right). \end{aligned} \quad (4)$$

With orthogonality in the basis vectors, the first term in the above equation can be factorized into

$$\boldsymbol{\mu}^\top\boldsymbol{\mu} = (\sum_i p_i\boldsymbol{e}_i)^\top(\sum_i p_i\boldsymbol{e}_i) = \sum_i p_i^2 \boldsymbol{e}_i^\top\boldsymbol{e}_i. \quad (5)$$

To encourage orthogonality in the basis vectors, a regularization term is added to the objective function:

$$\mathcal{L}_{\text{REG}}(\boldsymbol{x}; \boldsymbol{\phi}_1) = \|\boldsymbol{E}^\top\boldsymbol{E} - \alpha\boldsymbol{I}\|, \quad (6)$$

where $\boldsymbol{I}$ is the identity matrix and α is a hyperparamter. When $\mathcal{L}_{\text{REG}} = 0$, $\boldsymbol{e}_i^\top\boldsymbol{e}_i = \alpha$. In this case, $\boldsymbol{\mu}^\top\boldsymbol{\mu} = \alpha\sum_i p_i^2$ reaches its minimum $\frac{\alpha}{K}$ when $\boldsymbol{p}$ is a uniform distribution. The proof can be found in Appendix D. In practice, $\mathcal{L}_{\text{REG}}$ will quickly decrease to around 0, ensuring that the KL term will never fully collapse with the structural constraint. When it comes to controlled generation, one can choose a vertex or any desired point in the probability simplex.

4.3. Filling the Constrained Space

Constraining the posterior inside a certain space does not guarantee that this space will be filled after training. We also need to encourage the probability distribution over the relevant factors $\boldsymbol{p}$ to cover as much of the constrained latent space as possible. We introduce a reconstruction error of the structured latent code in order to push $\boldsymbol{p}$ away from a uniform distribution. For each input sentence, we randomly sample m sentences from the training data as negative samples. By applying the same encoding process, we get the structured latent code $\boldsymbol{\mu}_i^{(-)}$ for each negative sample. Our goal is to make the raw latent code $\boldsymbol{h}$ similar to the restructured latent code $\boldsymbol{\mu}$ while different from latent codes $\boldsymbol{\mu}_i^{(-)}$

of the negative samples, so that $\boldsymbol{p}$ is generally different for each input sample. The structured reconstruction loss is formulated as a margin loss as follows:

$$\mathcal{L}_{\text{S-REC}}(\boldsymbol{x};\boldsymbol{\phi}_1) = \mathbb{E}_{\boldsymbol{z}^{(1)}\sim q_{\boldsymbol{\phi}_1}(\boldsymbol{z}^{(1)}|\boldsymbol{x})}\left[\frac{1}{m}\sum_{i=1}^{m}\max(0, 1-\boldsymbol{h}\cdot\boldsymbol{\mu}+\boldsymbol{h}\cdot\boldsymbol{\mu}_i^{(-)})\right]. \tag{7}$$

Our final objective function is defined as follows:

$$\mathcal{L}(\boldsymbol{x};\boldsymbol{\theta},\boldsymbol{\phi}) = \mathcal{L}_{\text{VAE}} + \mathcal{L}_{\text{REG}} + \mathcal{L}_{\text{S-REC}}. \tag{8}$$

5. Experiments

To demonstrate the effectiveness of CP-VAE, we compare it to unsupervised baselines with β-VAE and state-of-the-art optimizing techniques, considering the performance on unsupervised sentiment manipulation. Following evaluation protocols in text style transfer, we also compare our method to strong supervised approaches. Furthermore, we showcase the ability of finer-grained style discovery and transition possessed by our system, which has not been attempted in the literature. Detailed configurations including the hyperparameters, model architecture, training regimes, and decoding strategy are found in Appendix C.

5.1. Comparisons with Unsupervised Baselines

Table 2. Comparisons with unsupervised baselines on Yelp dataset.

Model	Accuracy (AC) ↑	BLEU (BL) ↑
β-VAE ($\pm\sigma$)	50.98 ± 2.89	4.02 ± 0.77
β-VAE ($\pm 2*\sigma$)	78.44 ± 4.84	1.49 ± 0.29
β-VAE (extremum)	98.18 ± 1.56	0.56 ± 0.40
β-VAE w. aggr training ($\pm\sigma$)	26.76 ± 6.44	27.91 ± 4.39
β-VAE w. aggr training ($\pm 2*\sigma$)	57.46 ± 14.47	11.73 ± 6.74
β-VAE w. aggr training (extremum)	88.08 ± 14.95	4.57 ± 4.63
CP-VAE w. GloVe	60.22 ± 4.57	33.69 ± 1.47
without $\mathcal{L}_{\text{REG}}$	10.82 ± 0.91	33.27 ± 2.84
without $\mathcal{L}_{\text{S-REC}}$	12.28 ± 3.69	49.34 ± 2.65

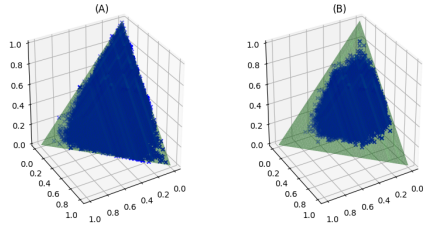

Figure 4. Visualization of all training samples in the probability simplex: (A) With $\mathcal{L}_{\text{S-REC}}$;(B) Without $\mathcal{L}_{\text{S-REC}}$.

Experimental setup: We use the same experimental setting and dataset as mentioned in Sec. 3. The 80D latent code is split into 16 and 64 dimensions for $\boldsymbol{z}^{(1)}$ and $\boldsymbol{z}^{(2)}$ respectively. The sentence representations for $\boldsymbol{z}^{(1)}$ is the averaged GloVe embeddings over the input tokens and K is chosen as 3. To decide which basis vector corresponds to which sentiment, we sample 10 positive and 10 negative sentences in the development set, pass them to the encoder, and choose the basis vector with the highest average p_i in $\boldsymbol{p} = \text{softmax}(\boldsymbol{W}\boldsymbol{h}+\boldsymbol{b})$, yielding v_p as the positive basis and v_n as the negative basis. If v_p and v_n are chosen to be the same vector, we choose the index with the second highest p_i for v_p. To perform sentiment manipulation, we fix $\boldsymbol{z}^{(1)}$ to be the chosen basis vector; that is, v_p or v_n.

Comparsions with metrics on text style transfer: For quantitative evaluation, we adopt two general automatic evaluation metrics used in text style transfer (Fu et al., 2018; Li et al., 2018; Sudhakar et al., 2019): classification accuracy (AC) of a pre-trained classifier to measure the transfer strength; BLEU score (BL) of the transferred sentences against the source sentences to measure the content preservation. As shown in Tab. 2, β-VAE alone performs poorly in terms of content preservation no matter the modification magnitude, while aggressively training the encoder can notably help improve content preservation. However, no matter we use aggressive training or not, the content preservation deteriorates drastically as we increase the modification magnitude, in order to achieve reasonable transfer strength. With large enough modification magnitude, the classification accuracy can be pushed to almost perfect, while the BLEU score decreases towards zero, meaning that the transferred sentences become totally irrelevant to the source sentences. The results match our observations from the experiments on density under the aggregated posterior distribution, confirming that latent vacancy prevents effective manipulation of the latent codes. To the contrary, CP-VAE can achieve much better content preservation while maintain its transfer strength, indicating its effectiveness to mitigate the latent vacancy problem.

Ablation study: We also conduct an ablation study by removing $\mathcal{L}_{\text{REG}}$ and $\mathcal{L}_{\text{S-REC}}$ from the objective. The results demonstrate that both terms are crucial to the success of CP-VAE. Without $\mathcal{L}_{\text{REG}}$, CP-VAE experiences posterior collapse for $\boldsymbol{z}^{(1)}$. As a result, v_p and v_n collide with each other, leading to failure in disentangled representation learning. Since we choose K as 3, it is convenient to visualize the samples during training with $\boldsymbol{p}$ in the learnt probability simplex, as shown in Fig. 4. We can see that the whole simplex is mostly covered with samples with the help of $\mathcal{L}_{\text{S-REC}}$. Without $\mathcal{L}_{\text{S-REC}}$, the decoding network fails to recognize the basis vectors due to the poor coverage of the probability simplex, causing the model to lose most of its transfer strength.

Table 3. Comparisons with supervised approaches on Yelp and Amazon dataset.

			Yelp				Amazon			
Model	Supervised	GPT-2	AC ↑	BL ↑	GL ↑	PL ↓	AC ↑	BL ↑	GL ↑	PL ↓
Source	-	-	1.8	100.0	8.4	26.6	16.3	100.0	22.8	34.5
Human	-	-	70.1	25.3	100.0	63.7	41.2	45.7	100.0	68.6
CA	✓	×	74.0	20.7	6.0	103.6	**75.5**	0.0	0.0	**39.3**
SE	✓	×	8.2	**67.4**	6.9	65.4	40.2	0.4	0.0	125.0
MD	✓	×	49.5	40.1	6.6	164.1	70.1	0.3	0.0	138.8
D&R	✓	×	**88.1**	36.7	7.9	85.5	49.2	0.6	0.0	46.3
CP-G	×	×	66.7	35.5	7.5	67.8	60.1	**35.4**	**11.5**	109.1
CP-B	×	×	55.4	48.4	**9.6**	**47.6**	40.0	**39.7**	**12.7**	97.3
B-GST	✓	✓	85.6	45.2	12.7	49.6	55.2	52.3	18.1	48.2

Table 4. Samples of generated sentences. SRC is the input sentence.

Yelp	*Positive to Negative*	*Negative to Positive*
SRC	this place is super yummy !	but it probably sucks too !
B-GST	this place is super bad !	but it tastes great too !
CP-G	this place is super slow and watered down .	but it 's truly fun and insanely delicious .
CP-B	this place is super greasy and gross !	but it 's probably wonderful when you !
Amazon	*Positive to Negative*	*Negative to Positive*
SRC	because it s made of cast iron , scorching is minimized .	they are cheerios, afterall, and we love the original kind .
B-GST	because it s cheaply made of cast iron , is useless .	they are sturdy, afterall, sturdy and we love the original .
CP-G	because it s made of cast iron , vomitting .	they are ripe, tastier , and we love them .
CP-B	because it s made of cast iron , limp .	they are divine, fluffier , and we love them .

At the same time, we do not claim that there are no other necessary conditions for the success of CP-VAE. First, if $\boldsymbol{z}^{(1)}$ uses raw text as inputs with a LSTM encoder, the VAEs will ignore $\boldsymbol{z}^{(1)}$ by making all the samples collapse to one vertex on the simplex. On the other hand, if $\boldsymbol{z}^{(2)}$ uses pre-trained embeddings with pooling like $\boldsymbol{z}^{(1)}$ as inputs, the VAEs would be unable to reconstruct the source sentence effectively, because the representations would lose most local information necessary for the reconstruction. However, this necessity is beside the point of our paper and does not contradict the evidence we presented for the latent vacancy hypothesis.

5.2. Comparisons to Supervised Approaches on Text Style Transfer

Experimental setup: We choose two datasets, Yelp and Amazon, used in works (Li et al., 2018; Sudhakar et al., 2019) on text style transfer which provide human gold-standard references for the test set. The same train-dev-test splits are used in our experiments. Two different sentence representations are used in this experiment, averaged GloVe and BERT (Devlin et al., 2018), denoted as **CP-G(loVe)** and **CP-B(ert)** respectively. The remaining settings are as described in the above section.

Compared supervised approaches: On the two datasets, we compare to three adversarially trained models: StyleEmbedding (**SE**) (Fu et al., 2018), MultiDecoder (**MD**) (Fu et al., 2018), CrossAligned (**CA**) (Shen et al., 2017) and two state-of-the-art models based on a "delete, transform, and generate" framework: DeleteAndRetrieve (**D&R**) (Li et al., 2018) and Blind-GenerativeStyleTransformer (**B-GST**) (Sudhakar et al., 2019). To be noted, the decoding network of **B-GST** is based on GPT-2 (Radford et al., 2019), while all the other models including ours train the decoding network from scratch.

Evaluation protocols: Four different automatic evaluation metrics are used to measure the different perspectives of the transferring quality, following Sudhakar et al. (2019). To measure transfer strength, we use pre-trained CNN based classifiers achieving 98% and 84% accuracies on the test sets of Yelp and Amazon respectively. To measure content preservation, we use the BLEU (Papineni et al., 2002) score of the transferred sentences against the source sentences. To measure fluency, we finetune OpenAI GPT-2 (Radford et al., 2019) with 345 million parameters on the same training-dev-test split to obtain the perplexity of generated sentences. The fine-tuned language models achieve perplexities of 26.6 and 34.5 on the test sets of Yelp and Amazon respectively. In addition, Sudhakar et al. (2019) argued that the Generalized Language Evaluation Understanding Metric (GLEU) has a better correlation with the human judgement. Here,

Table 5. Two pairs of samples generated without and with topic transition. The first sentence in the pair is generated with a topic fixed throughout the generation; while the second sentence is generated with topic transition, the generated outputs after switching are marked as bold.

World throughout	A federal judge on Friday ordered a federal appeals court to overturn a federal appeals court ruling that the Visa and MasterCard credit card associations violated federal antitrust law by barring the names of the state .
World to *Sci/Tech*	A federal judge on Friday ordered a federal appeals court to overturn a decision by the Supreme Court to **overturn a decision by the Federal Communications Commission to block the company's antitrust case against Microsoft Corp .**
Sports throughout	NEW YORK (Reuters) - Roger Federer, the world's No. 1 player, will miss the rest of the season because of a sore quadriceps .
Sports to *Business*	NEW YORK (Reuters) - Roger Federer, the world's No. 1 player, will miss the rest of the **year because of a bid-rigging scandal .**

we use the implementation of GLEU[4] provided by Napoles et al. (2015) to calculate the GLEU score.

Result Analysis: As observed by Li et al. (2018) and Sudhakar et al. (2019), accuracy, BLEU score and perplexity do not correlate well with human evaluations. Therefore, it is important to not consider them in isolation. Tab. 3 shows that our proposed approaches get similar scores on these metrics with human reference sentences on the second row, indicating that the generated sentences of our proposed approaches is reasonable considering the combination of these metrics. As seen by Sudhakar et al. (2019) and verified in Sec. 5.1, GLEU strike a balance between target style match and content retention and correlate well with the human evaluations. From Tab. 3, CP-VAE consistently outperforms the three adversarially trained models and D&R on GLEU by a noticeable margin. As compared to B-GST, the current state-of-the-art, which leverages GPT-2 for generation, the results are still competitive, despite the fact that CP-VAE is trained unsupervisedly and from scratch. By checking the samples generated from the models as shown in Tab. 4, B-GST is more consistent to the source sentence, which can be expected, since it only makes necessary edits to flip the sentiment. CP-VAE tends to generate more diverse contents which may not be relevant sometimes, but the overall quality is reasonable. More samples can be found in Appendix F.

5.3. Finer-grained Style Discovery and Transition

To further explore the potential of CP-VAE, we conduct the following exploratory experiments. We use the AG news dataset constructed by (Zhang et al., 2015), which contains four topic categories which are *World, Sports, Business* and *Sci/Tech*, with the title and description fields. Here, we drop the title and just use the description field to train CP-VAE and set $K = 10$. All four topics are automatically discovered by CP-VAE and identified as described in Sec. 5.1. We also compare the results of our identified topics to standard baselines for unsupervised topic modelling, the details can be found in Appendix E. We choose a basis vector discovered by our model and generate a few tokens. Then, we switch the basis vector and continue the generation until the *end-of-seq* token is generated. Generated samples are shown in Table 5. We see that our model learns to transition from one topic to another in a natural and fluent way within the same sentence. Several observations can be made based on these samples: (1) it is good at detecting name entities and replacing them with the name entities related to the chosen topic; (2) there is no hard restriction on when to switch the topic; the model will determine an appropriate way to do the transition by itself. Such observations confirm that CP-VAE possesses a filled constrained latent space which make the latent code robust to manipulation across different time steps, which can be effectively reflected in the generation process. Due to space limitations, we put more samples in Appendix G.

[4]https://github.com/cnap/gec-ranking

6. Related Work

6.1. Unsupervised Learning of Disentangled Representations

Learning disentangled representations is an important step towards better representation learning (Bengio et al., 2013) which can be useful for (semi-)supervised learning of downstream tasks, transfer and few-shot learning (Peters et al., 2017). VAEs have achieved promising results for unsupervised learning of disentangled representations. Several variations of VAEs have been proposed for better disentanglement (Higgins et al., 2017; Kumar et al., 2017; Chen et al., 2016; Razavi et al., 2019). However, progress in this direction has been restricted to the image domain, and does not demonstrate successful controlled generation on text.

6.2. Controlled Text Generation

In order to perform controllable text generation, previous methods either assume annotated attributes or multiple text datasets with different known styles (Hu et al., 2017; Shen et al., 2017; Zhao et al., 2018; Fu et al., 2018; Li et al., 2018;

Sudhakar et al., 2019; Logeswaran et al., 2018; Lample et al., 2018). The requirement of labelled data largely restricts the capabilities and the applications of these models. Instead, all our proposed framework needs is raw text without any annotated attribute.

7. Conclusion

In this work, we investigate latent vacancy as an important problem in unsupervised learning of controllable representations when modelling text with VAEs. To mitigate this, we propose to constrain the posterior within a learned probability simplex and encourage this space to be filled, achieving the first success towards controlled text generation without supervision. However, the constrained posterior also means that the aggregated posterior can never match the isotropic Gaussian prior which points to a potential future direction to resolve this mismatch by selecting or learning a better prior as in (Tomczak & Welling, 2017).

Acknowledgements

Thanks to Ivan Kobyzev for the useful discussion and feedback, and to all the anonymous reviewers for their valuable inputs.

References

Bao, Y., Zhou, H., Huang, S., Li, L., Mou, L., Vechtomova, O., Dai, X., and Chen, J. Generating sentences from disentangled syntactic and semantic spaces. *arXiv preprint arXiv:1907.05789*, 2019.

Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8): 1798–1828, 2013.

Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan): 993–1022, 2003.

Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., and Bengio, S. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pp. 2172–2180, 2016.

Cífka, O., Severyn, A., Alfonseca, E., and Filippova, K. Eval all, trust a few, do wrong to none: Comparing sentence generation models. *arXiv preprint arXiv:1804.07972*, 2018.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Fu, Z., Tan, X., Peng, N., Zhao, D., and Yan, R. Style transfer in text: Exploration and evaluation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

He, J., Spokoyny, D., Neubig, G., and Berg-Kirkpatrick, T. Lagging inference networks and posterior collapse in variational autoencoders. *arXiv preprint arXiv:1901.05534*, 2019.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, volume 3, 2017.

Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., and Xing, E. P. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1587–1596. JMLR. org, 2017.

Hyndman, R. J. Computing and graphing highest density regions. *The American Statistician*, 50(2):120–126, 1996.

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

Kumar, A., Sattigeri, P., and Balakrishnan, A. Variational inference of disentangled latent concepts from unlabeled observations. *arXiv preprint arXiv:1711.00848*, 2017.

Lample, G., Subramanian, S., Smith, E., Denoyer, L., Ranzato, M., and Boureau, Y.-L. Multiple-attribute text rewriting. 2018.

Li, J., Jia, R., He, H., and Liang, P. Delete, retrieve, generate: A simple approach to sentiment and style transfer. *arXiv preprint arXiv:1804.06437*, 2018.

Logeswaran, L., Lee, H., and Bengio, S. Content preserving text generation with attribute controls. In *Advances in Neural Information Processing Systems*, pp. 5103–5113, 2018.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

Manning, C., Raghavan, P., and Schütze, H. Introduction to information retrieval. *Natural Language Engineering*, 16 (1):100–103, 2010.

Napoles, C., Sakaguchi, K., Post, M., and Tetreault, J. Ground truth for grammatical error correction metrics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 588–593, 2015.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 311–318. Association for Computational Linguistics, 2002.

Pennington, J., Socher, R., and Manning, C. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

Peters, J., Janzing, D., and Schölkopf, B. *Elements of causal inference: foundations and learning algorithms*. MIT press, 2017.

Radford, A., Jozefowicz, R., and Sutskever, I. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444*, 2017.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog*, 1:8, 2019.

Razavi, A., Oord, A. v. d., Poole, B., and Vinyals, O. Preventing posterior collapse with delta-vaes. *arXiv preprint arXiv:1901.03416*, 2019.

Rezende, D. J. and Viola, F. Taming vaes. *arXiv preprint arXiv:1810.00597*, 2018.

Shen, T., Lei, T., Barzilay, R., and Jaakkola, T. Style transfer from non-parallel text by cross-aligment. *Advances in neural information processing systems*, pp. 6830–6841, 2017.

Singh, G., Mémoli, F., and Carlsson, G. E. Topological methods for the analysis of high dimensional data sets and 3d object recognition. 2007.

Sudhakar, A., Upadhyay, B., and Maheswaran, A. Transforming delete, retrieve, generate approach for controlled text style transfer. *arXiv preprint arXiv:1908.09368*, 2019.

Tomczak, J. M. and Welling, M. Vae with a vampprior. *arXiv preprint arXiv:1705.07120*, 2017.

van Veen, H., Saul, N., Eargle, D., and Mangham, S. Kepler mapper: A flexible python implementation of the mapper algorithm. *Journal of Open Source Software*, 4(42):1315, 2019.

Wen, T.-H., Vandyke, D., Mrksic, N., Gasic, M., Rojas-Barahona, L. M., Su, P.-H., Ultes, S., and Young, S. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*, 2016.

Zhang, X., Zhao, J., and LeCun, Y. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pp. 649–657, 2015.

Zhao, J., Kim, Y., Zhang, K., Rush, A. M., LeCun, Y., et al. Adversarially regularized autoencoders. *Proceedings of the 35th International Conference on Machine Learning*, 2018.