Variational Autoencoder with Disentanglement Priors for Low-Resource Task-Specific Natural Language Generation

Zhuang Li[△], **Lizhen Qu**[*,△], **Qiongkai Xu**[♡]
Tongtong Wu[△], **Tianyang Zhan**[†,♢], **Gholamreza Haffari**[△]
Monash University, Australia[△,♢], The University of Melbourne, Australia[♡]
firstname.lastname@monash.edu[△], firstname.lastname@unimelb.edu.au[♡],
tzha225@student.monash.edu[♢]

Abstract

In this paper, we propose a variational autoencoder with disentanglement priors, VAE-DPRIOR, for task-specific natural language generation with none or a handful of task-specific labeled examples. In order to tackle compositional generalization across tasks, our model performs disentangled representation learning by introducing a conditional prior for the latent content space and another conditional prior for the latent label space. Both types of priors satisfy a novel property called ϵ-disentangled. We show both empirically and theoretically that the novel priors can disentangle representations even without specific regularizations as in the prior work. The content prior enables directly sampling diverse content representations from the content space learned from the seen tasks, and fuse them with the representations of novel tasks for generating semantically diverse texts in the low-resource settings. Our extensive experiments demonstrate the superior performance of our model over competitive baselines in terms of i) data augmentation in continuous zero/few-shot learning, and ii) text style transfer in the few-shot setting. The code is available at https://github.com/zhuang-li/VAE-DPrior.

1 Introduction

Task-specific Natural Language Generation (NLG) aims to generate texts that satisfy desired attributes of target tasks, such as text style transfer (Jin et al., 2020) and task-specific data augmentation (Lee et al., 2021). Herein, a task includes a set of task-specific labels, optionally a set of labeled texts for that task (Han et al., 2020). Although there is already a large amount of labeled data for various tasks, in many application scenarios, such as AI assistants for legal aid, the labeled data of new tasks are still difficult to acquire. As a result, there may be no or just a handful of labeled texts for target tasks. In such a low-resource setting, given a new task, it is desirable to i) identify which information in texts is task-specific and which is task-independent, and ii) systematically and consistently combine the label representations of the new task with task-independent content representations for text generation. As illustrated in Fig. 1, data augmentation needs to combine content representations from seen tasks with novel task labels. In contrast, text style transfer requires combining the content representations extracted from inputs with target styles.

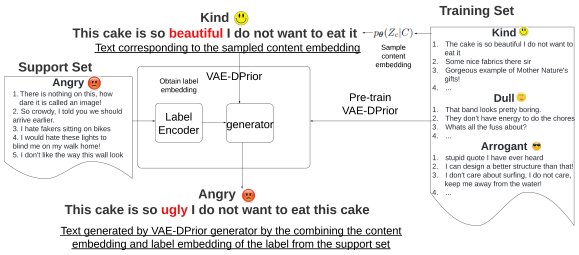

Figure 1: Generation of task-specific examples for data augmentation. In this example, the content representation is sampled from the training set, while the label representation is constructed based on the support set.

Most prior work assumes access to labeled data for supervised training. However, those models trained on seen tasks cannot generalize well to new tasks during inference (Krishna et al., 2022). One of the key reasons is that the parameters of supervised models are tied to seen tasks such that a significant amount of fine-tuning data is needed for adapting to new tasks. For prompt-based and guided decoding methods (Zhang et al., 2022), although they require significantly less training data, it is still challenging to generate a large number of semantically diverse and coherent texts for new tasks in a robust way because they cannot well leverage the rich contents of the seen tasks.

The key challenge of low-resource task-specific NLG is to disentangle content representations from

[*] corresponding author
[†] Most of this author's work was finished when he was with Monash University.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 10335 - 10356
December 7-11, 2022

label representations with few labeled data of target tasks. If content representations still contain task-specific information from seen tasks, they may well mislead the language generator after fusing with the representations of new tasks. Prior works tackle this problem by enforcing the random variables of content representations to be independent of those of label representations (Cheng et al., 2020). However, in practice, both types of random variables are not always independent. For example, the random variables of emotion labels naturally depend on the contents of the events causing them.

In this work, we propose a deep VAE model with *novel* disentanglement priors, coined VAE-DPRIOR, for task-specific natural language generation in the zero-shot and few-shot settings. In contrast to the widely used *unconditional* priors in the VAE framework, the new priors are *conditional*, satisfying a *novel* property called ϵ-disentangled, which motivates a new way of regularization for disentangling representations *without forcing independence* between the corresponding random variables. The new priors build a constraint space for latent content representations and latent label representations with the aims to i) minimize information overlap between the two types of representations and ii) enable generalization across tasks with little labeled training data. One of the priors is a *conditional* Gaussian mixture in the content subspace for sampling rich content representations without accessing original training data. Another type of priors is a *conditional* multivariate Gaussian per label that associates latent label representations with task-specific information, requiring only a label name or a small set of labeled examples. Extending a pre-trained language decoder based on the prefix-tuning technique (Li and Liang, 2021) with those priors, our model is able to sample rich content representations of seen labels and combine them with the representations of new labels to generate *diverse* and *natural* sentences. In addition, we empirically observe that VAE-DPRIOR alleviates posterior collapse (Wang et al., 2020), which is a long-standing problem of VAEs that makes it difficult to train a latent model to generate coherent and semantically diverse texts.

To sum up, our key contributions are threefold: i) We propose a VAE-DPRIOR model with *novel* disentanglement priors for low-resource task-specific NLG tasks. It enables sampling diverse content representations directly from the content

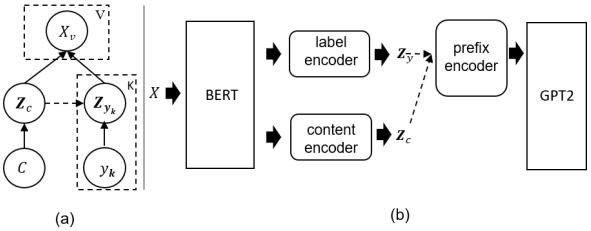

Figure 2: (a) A directed graphical model for disentanglement learning. (b) The architecture of VAE-DPRIOR.

prior; ii) We introduce ϵ-disentangled, which sets a *novel* regularization goal for disentangled representations; iii) Our model outperforms competitive baselines in the low-resource settings on the tasks of text style transfer and data augmentation for continual few/zero shot text classification.

2 Methodology

To tackle task-specific NLG tasks in low-resource settings, we introduce a deep generative model VAE-DPRIOR, which employs disentanglement priors, including a content prior for rich contents, to generate coherent and semantically diverse texts. We are provided with a large corpus of labeled sentences $\mathcal{D}^{(0)} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{n}$ for an initial task $\mathcal{T}^{(0)}$, where a sentence $\boldsymbol{x}_i \in \mathcal{X}$ is annotated with a seen label $y_i \in \mathcal{Y}$. The goal is to learn a single model that can generate diverse texts for any new task or a sequence of K distinct new tasks $\{\mathcal{T}^{(1)}, \mathcal{T}^{(2)},...,\mathcal{T}^{(K)}\}$. Each new task includes multiple novel labels, where a label $y \in \mathcal{Y}$ is associated with a label name and optionally a handful of example texts $\mathcal{D}_{sup} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{m}$ as the *support* set. The model is evaluated on both *data augmentation for continual text classification* described in Sec. 3 and *few-shot text style transfer* detailed in Appendix C.1.

For evaluating data augmentation, a text classifier is trained *sequentially* on K new tasks and evaluated on the test sets of all seen tasks $\{\mathcal{T}^{(1)},...,\mathcal{T}^{(t)}\}$ till time t. For each task, its training data includes the texts generated by the NLG models in order to evaluate to what degree the augmented texts improve the classifier performance. In the zero-shot setting, the classifier is trained only on the generated texts using the label names of new tasks, while the support sets are also used for data augmentation and classifier training in the few-shot setting.

2.1 Theoretical Framework

In the absence of large training data for new tasks, one of the key challenges is to construct content

representations and label representations in the latent space satisfying *information purity*. As such, content representations should not contain label information, otherwise old task information in such content representations may contaminate combined representations for new tasks, vice versa.

Formally, the latent space is the sample space Ω for both content and label representations. In the corresponding probability space, we define a random variable vector $\mathbf{Z}_y$ for the latent representations of each label $y \in \mathcal{Y}$, a random variable vector $\mathbf{Z}_c$ for latent content representations. Then observable word sequences are denoted by the random variable vector $\mathbf{X}$, where each variable X_v corresponds to a word in the vocabulary $\mathcal{V}$. The statistical dependencies between those random variables are illustrated by the Bayesian network in Fig.2(a), where C denotes the prior knowledge of contents. The dashed arrow denotes a possible dependency between $\mathbf{Z}_y$ and $\mathbf{Z}_c$.

To achieve information purity, the learned models are expected to follow the structure illustrated in Fig.2(a) that there is no dependency between C and $\mathbf{Z}_y$, as well as no dependency between y and $\mathbf{Z}_c$. However, prior works on disentangled representation learning regularize the models by approximating $\mathbf{Z}_c \perp\!\!\!\perp \mathbf{Z}_y$ (Cheng et al., 2020; Wang and Jordan, 2021), which may violate the true statistical relation between $\mathbf{Z}_c$ and $\mathbf{Z}_y$. Even though $\mathbf{Z}_c \perp\!\!\!\perp \mathbf{Z}_y$ holds after regularization, it does not imply $\mathbf{Z}_y \perp\!\!\!\perp C$ and $\mathbf{Z}_c \perp\!\!\!\perp y$. The random variable of a label can still depend on both $\mathbf{Z}_c$ and $\mathbf{Z}_y$.

To address this limitation, we propose to regularize the priors of the latent variables for encouraging information purity. After training, we expect the mutual information of $I(\mathbf{Z}_y, Y)$ and $I(\mathbf{Z}_c, C)$ is high, while $I(\mathbf{Z}_y, C)$ and $I(\mathbf{Z}_c, y)$ is low or zero. One way to achieve this is that we make $I(\mathbf{Z}_y, y)$ high only in the dense regions of $\mathbf{Z}_y$ but force it to be low or zero in the dense regions of $\mathbf{Z}_c$, vice versa. As a result, we expect little overlap between the dense regions of $p_{\boldsymbol{\theta}_c}(\mathbf{Z}_c|C)$ and those of $p_{\boldsymbol{\theta}_y}(\mathbf{Z}_y|y)$. Then the distances between those priors are large. We characterize this property by introducing ϵ-disentangled below.

Definition 2.1 (ϵ-disentangled)**.** Two distributions $p_{\boldsymbol{\theta}_c}(\mathbf{Z}_c|C)$ and $p_{\boldsymbol{\theta}_y}(\mathbf{Z}_y|y)$ are ϵ-disentangled, if $1/\mathbb{D}_k(p_{\boldsymbol{\theta}_c}(\mathbf{Z}_c|C)||p_{\boldsymbol{\theta}_y}(\mathbf{Z}_y|y)) \leq \epsilon$ and $\epsilon \in \mathbb{R}^+$, where $\mathbb{D}_k$ denotes a divergence measure requiring no absolute continuity (Royden and Fitzpatrick, 1988), then $p_{\boldsymbol{\theta}_c}(\mathbf{Z}_c|C)$ and $p_{\boldsymbol{\theta}_y}(\mathbf{Z}_y|y)$ are ϵ-disentangled w.r.t. the measure $\mathbb{D}_k$.

We refer to the priors satisfying ϵ-disentangled as *disentanglement priors*. In Appendix E.1, we conduct an in-depth discussion of this property. We show that if $p_{\boldsymbol{\theta}_c}(\mathbf{Z}_c|C)$ and $p_{\boldsymbol{\theta}_y}(\mathbf{Z}_y|y)$ are not ϵ-disentangled under a mild assumption, at least one of them is non-identifiable, which is a leading cause of posterior collapse (Wang et al., 2020).

VAE with disentanglement priors. Using the disentanglement priors, we employ the maximum likelihood principle for learning the parameters of the joint distribution $\prod_{y\in\mathcal{Y}} p_{\boldsymbol{\theta}}(\mathbf{X}, \mathbf{Z}_c, \mathbf{Z}_y|C, y)$. The marginal distribution $\prod_{y\in\mathcal{Y}} p_{\boldsymbol{\theta}}(\mathbf{X}|C, y)$ is given by

$$\int \prod_{y\in\mathcal{Y}} p_{\boldsymbol{\theta}}(\mathbf{X}|\mathbf{Z}_c, \mathbf{Z}_y, y, C) p_{\boldsymbol{\theta}_y}(\mathbf{Z}_y|y) p_{\boldsymbol{\theta}_c}(\mathbf{Z}_c|C) d\mathbf{Z}_y d\mathbf{Z}_c. \tag{1}$$

We learn the above distribution in the VAE framework. Note that, the introduction of the conditions C and y makes the priors of both latent variables *conditional*, which differs from vanilla VAEs that have only *unconditional* priors for latent variables.

Given a dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, the training problem to learn the marginal distribution in Eq. 1 is formulated as:

$$\max \sum_{i=1}^{n} \log p(\boldsymbol{x}_i|C, y_i)$$
$$s.t.\ p_{\boldsymbol{\theta}_c}(\mathbf{Z}_c|C) \text{ and } p_{\boldsymbol{\theta}_y}(\mathbf{Z}_y|y) \text{ are } \epsilon\text{-disentangled.} \tag{2}$$

The disentanglement constraint is achieved by either carefully choosing priors satisfying ϵ-disentangled, applying a divergence measure requiring no absolute continuity between priors as a regularizer, such as the Maximum Mean Discrepancy (MMD) (Gretton et al., 2012), or both. In the following, we provide the model details and show how to derive an evidence lower bound (ELBO) in the VAE framework for this optimization problem.

2.2 Model Details

As illustrated in Fig. 2(b), the overall architecture consists of an inference module, a generator and priors. The inference module consists of a pre-trained BERT encoder, whose outputs serve as inputs of a label encoder and a content encoder, and a generator comprising a prefix encoder and a pre-trained GPT2 with frozen parameters.

The VAE framework adopts variational distributions to approximate true distributions (Kingma and Welling, 2019), which ends up maximizing an

ELBO. We show in Appendix E.3 that the ELBO objective takes the following form:

$$\begin{aligned}&\overbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{Z}_c,\mathbf{Z}_y|\mathbf{X},C,y)}[\log p_{\boldsymbol{\theta}}(\mathbf{X}|\mathbf{Z}_c,\mathbf{Z}_y)]}^{\mathcal{L}_r}\\&-\mathbb{D}_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{Z}_c|\mathbf{X},C)\|p_{\boldsymbol{\theta}_c}(\mathbf{Z}_c|C))\\&-\mathbb{D}_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{Z}_y|\mathbf{X},y)\|p_{\boldsymbol{\theta}_y}(\mathbf{Z}_y|y)),\end{aligned} \tag{3}$$

where the first term is referred to as the reconstruction loss $\mathcal{L}_r$, the other terms constitute regularizers. Following the convention of VAE, we refer to the network for $q_{\boldsymbol{\phi}}(\mathbf{Z}_c, \mathbf{Z}_y|\mathbf{X}, C, y)$ as *inference module*, the network for $p_{\boldsymbol{\theta}}(\mathbf{X}|\mathbf{Z}_c, \mathbf{Z}_y)$ as *generator*.

Priors. In the label subspace, we assume $p_{\boldsymbol{\theta}_y}(\mathbf{Z}_y|y)$ for a label y is a simple factorized Gaussian distribution in form of $\mathcal{N}(\mathbf{Z}_y; \boldsymbol{\mu}_y^p, \lambda_y\mathbf{I})$, where λ_y is a hyperparameter, its mean $\boldsymbol{\mu}_y^p$ is constructed by using the name embedding of label y in the zero-shot setting, and by averaging the label name embedding and the embeddings of its support set examples in the few-shot setting. Each embedding is curated by feeding its word sequence to the label encoder shared with that of the inference module.

The content prior $p_{\boldsymbol{\theta}_c}(\mathbf{Z}_c|C)$ takes the form of $\sum_{k=1}^{K} p_{\boldsymbol{\theta}}(M = k)\mathcal{N}(\mathbf{Z}_c; \boldsymbol{\mu}_{c,k}^p, \lambda_c\mathbf{I})$, where M is the random variable indicating the membership to a component Gaussian. Inspired by neural topic modelling (Wang and Yang, 2020), we encode the prior knowledge of content C into a k-means clusters, where we assume that there is a one-to-one correspondence between a component Gaussian and a cluster in the k-means clusters. The mean of a Gaussian component $\mathcal{N}(\mathbf{Z}_c; \boldsymbol{\mu}_{c,k}^p, \lambda_c\mathbf{I})$ is computed by $\mathbf{W}_c\boldsymbol{c}_k$, a linear projection from the corresponding cluster centroid $\boldsymbol{c}_k$. The k-means clusters are built from BERT sentence embeddings on the training data of seen tasks. Adding new topics is a matter of adding new clusters using incremental clustering techniques.

In Appendix E.2, we show that $p_{\boldsymbol{\theta}_c}(\mathbf{Z}_c|C)$ and $p_{\boldsymbol{\theta}_y}(\mathbf{Z}_y|y)$ are ϵ-disentangled with a small ϵ if their means are far from each other and their variances are sufficiently small.

Inference Module. The inference module is a BERT (Devlin et al., 2018) encoder augmented with an encoder for content, and an encoder for labels. Each encoder is built on top of the contextual embedding sequences produced by BERT and yields latent representations of the target type. This design is not only parameter efficient but also leverages the strengths of a large-scale pre-trained transformer model.

A BERT model consists of multiple layers. To provide more model capacities to capture the differences between the two types of latent representations and preserve parameter efficiency, we freeze all layers of BERT except the top most one so that the content encoder and the label encoder employ a top most transformer layer with different parameters respectively, while sharing all the remaining layers of BERT.

In the label subspace, given a contextual word embedding sequence $\mathbf{V}_l = \{\boldsymbol{v}_0, ..., \boldsymbol{v}_u\}$ generated by the corresponding top-most layer of BERT, the **LabelEncoder** implements $q_{\boldsymbol{\phi}}(\mathbf{Z}_y|\mathbf{X}, y)$ in form of $\mathcal{N}(\mathbf{Z}_y; \boldsymbol{\mu}_y^q, \mathrm{diag}(\boldsymbol{\sigma}_y^2))$. In order to build a hidden representation focusing on label relevant information, we apply the label embedding $\boldsymbol{\mu}_y^p$ used in the label prior to $\mathbf{V}_y$ via soft attention. In particular, we compute an aggregated representation $\boldsymbol{h}_y = \mathrm{attention}(\boldsymbol{\mu}_y^p, \mathbf{V}_l)$ for a label y by applying $\boldsymbol{\mu}_y^p$ as the query vector to attend all vectors of $\mathbf{V}_y$. We compute the mean $\boldsymbol{\mu}_y^q$ as a linear transformation of $\boldsymbol{h}_y$ by using the weight matrix $\mathbf{W}_\mu^l$ and the logarithm of the variance $\log \boldsymbol{\sigma}_y$ as the linear transformation of another linear matrix $\mathbf{W}_\sigma^l$.

By applying the reparameterization trick (Kingma and Welling, 2019), the latent label representation $\boldsymbol{z}_y$ is a function of $\boldsymbol{\mu}_y^q$ and a stochastic noise. The stochastic noise is added by the product of $\boldsymbol{\sigma}_y$ and the Gaussian noise $\boldsymbol{\epsilon}_y$ drawn from $\mathcal{N}(0, \mathbf{I})$.

$$\begin{aligned}\log \boldsymbol{\sigma}_y, \boldsymbol{\mu}_y^q &= \mathrm{LabelEncoder}(\mathbf{V}_l)\\ \boldsymbol{z}_y &= \boldsymbol{\mu}_y^q + \boldsymbol{\sigma}_y \odot \boldsymbol{\epsilon}_y\end{aligned} \tag{4}$$

where $\odot$ denotes the element-wise product.

In the content subspace, we consider $q_{\boldsymbol{\phi}}(\mathbf{Z}_c|\mathbf{X}, C) = \mathcal{N}(\mathbf{Z}_c; \boldsymbol{\mu}_c^q, \mathrm{diag}(\boldsymbol{\sigma}_c^2))$. Taking $\mathbf{V}_c$ from the corresponding top most layer of BERT as input, **ContentEncoder** consists of a mean pooling layer followed by a linear layer for the mean and another linear layer for the logarithm of the variance of $q_{\boldsymbol{\phi}}(\mathbf{Z}_c|\mathbf{X}, C)$. The same reparameterization trick is applied to obtain the latent representation $\boldsymbol{z}_c$.

Generator. Given a pair of latent representations $(\boldsymbol{z}_c, \boldsymbol{z}_y)$, the generator captures $p(\mathbf{X}|\boldsymbol{z}_c, \boldsymbol{z}_y)$ factorized into the following autoregressive form.

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}_c, \boldsymbol{z}_y) = \prod_{t=1}^{|\boldsymbol{x}|} p_{\boldsymbol{\theta}}(x_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_c, \boldsymbol{z}_y) \tag{5}$$

We employ a prefix-tuning technique (Li and Liang, 2021) that yields continuous prompts for the decoder in the low-resource situations. A continuous prompt is a continuous vector sequence of length L. The *prefix encoder* consists of L MLPs,

each of which uses the architecture $\boldsymbol{M}_\theta[i,:] = \text{MLP}_\theta([\boldsymbol{M}'_\theta[i,:]; \boldsymbol{z}_y; \boldsymbol{z}_c])$ for computing a vector at position i, where $\boldsymbol{M}'_\theta \in \mathbb{R}^{|M_{idx}| \times H'}$ is a learned matrix to encode the position information of the continuous prompt. In practice, we find it also useful to prepend the name embedding of the target label to the prefix. Further implementation details are available in Appendix A.

In the low-resource settings, the mechanisms of constructing latent label and content representations across tasks should be consistent, otherwise labeled data needs to be provided for adjusting model parameters for alleviating those discrepancies. Therefore, we minimize parameters to update across tasks, use the same prior for content representations, and construct label embeddings using the same label encoder for different tasks. The parameters of the pre-trained encoder and the pre-trained decoder are frozen during training. Thus, we only need to train the parameters of the intermediate hidden layers between them on the data of initial tasks, which are also frozen for new tasks. Freezing parameters could effectively prevent the catastrophic forgetting of models when learning the new tasks.

2.3 Training and Inference

Given a training corpus $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^n$, we derive the objective function $\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathcal{D}) = \mathcal{L}_r + \sum_y \mathcal{L}_y(y) + \mathcal{L}_c$ from the objective in Eq. 2 and the ELBO, where $\mathcal{L}_y$ and $\mathcal{L}_c$ are the KL regularization terms from the ELBO. The constraint is removed by the usage of the disentanglement priors.

We first pre-train the whole model on the corpus of the initial task $\mathcal{T}^{(0)}$ without applying any disentanglement constraint and the regularizers derived from the ELBO, followed by fine tuning the model with all regularizers. In practice, we find out that the two-steps approach is important to achieve optimal empirical performance.

Regularization in the Label Subspace. The regularization term $\mathcal{L}_y(y)$ in the label subspace is derived from $\mathbb{D}_{\text{KL}}(q_{\boldsymbol{\phi}}(\mathbf{Z}_y|X,y)\|p_{\boldsymbol{\theta}_y}(\mathbf{Z}_y|y))$.

$$\mathcal{L}_y(y) = -\frac{1}{\lambda_y}\|\boldsymbol{z}_y - \boldsymbol{\mu}_y^p\|^2 + \log \boldsymbol{\sigma}_y^q \tag{6}$$

The first term enforces latent label representations $\boldsymbol{z}_y$ to be close to the label prototype $\boldsymbol{\mu}_y^p$ obtained from the label priors. In contrast, the corresponding regularization term in a vanilla VAE with unconditional Gaussian priors takes the form of $\|\boldsymbol{z}_y\|^2$, which only makes the latent representations smooth without providing any label specific information.

Regularization in the Content Subspace. Derived from $\mathbb{D}_{\text{KL}}(q_{\boldsymbol{\phi}}(\mathbf{Z}_c|X,C)\|p_{\boldsymbol{\theta}_c}(\mathbf{Z}_c|C))$, the regularization term $\mathcal{L}_c$ takes the similar form as the loss of deep k-means (Fard et al., 2020).

$$\mathcal{L}_c = \sum_{k=1}^{K} p(M=k|\boldsymbol{z}_c)\big[-\frac{1}{2\lambda_c}\|\boldsymbol{z}_c - \boldsymbol{\mu}_{c,k}^p\|^2\big] + \log \boldsymbol{\sigma}_c \tag{7}$$

We compute it by using EM. The term $q_{\boldsymbol{\phi}}(M_k|\boldsymbol{x})$ is computed by the E-step. If soft-EM is considered, $q_{\boldsymbol{\phi}}(M_k|\boldsymbol{x}) = \frac{\exp(-\text{dist}(\boldsymbol{z}_c, \boldsymbol{\mu}_{c,k}^p)/\tau)}{\sum_{k'}\exp(-\text{dist}(\boldsymbol{z}_c, \boldsymbol{\mu}_{c,k'}^p)/\tau)}$, which denotes the probability of an example $\boldsymbol{x}$ belonging to a cluster k with a temperature τ. In our experiments, we employ hard EM, where $q_{\boldsymbol{\phi}}(M_k|\boldsymbol{x})$ indicates if the current Gaussian has the same the index as the one having the minimal Euclidean distance $\|\boldsymbol{z}_c - \boldsymbol{\mu}_{c,k}^p\|^2$ among all components.

Inference. For data augmentation, our model samples a large number of texts from the model and filters out the ones that are not in accordance with the target labels. For each new label y, we construct the mean $\boldsymbol{\mu}_y^p$ of $p_{\boldsymbol{\theta}_y}(\mathbf{Z}_y|y)$ by averaging the embeddings of the label name phrase and optionally its associated texts from the support set. The corresponding embeddings are generated by feeding name phrases and texts into the label encoder. Then we sample a large number of content embeddings from the content prior $p_{\boldsymbol{\theta}}(\mathbf{Z}_c|C)$. All combinations of label embeddings and content embeddings are fed to the generator to generate candidate examples. We find that the candidates of low quality are not in accordance with the target labels. Hence, we perform **quality control** by filtering out irrelevant ones. Specifically, we project each candidate to a latent representation using the label encoder, and rank all candidates w.r.t. the Euclidean distance between each representation and its associated name embedding. The top-k candidates are taken as the final outputs.

3 Experiments

We evaluate our model on both continual few/zero shot text classification and few-shot text style transfer. The former requires sampling rich content representations from seen tasks, while the latter expects to retain task-independent contents from inputs. In both cases, it is desirable for models to systematically combine latent label and content representations across tasks in a consistent manner.

The details of **few-shot text style transfer** are available in Appendix C. We compare VAE-DPRIOR with five style-transfer baselines and show

superior results on two datasets in the few-shot setting in terms of accuracy of style transfer, semantic relevance and naturalness of the generated text.

3.1 Continual Zero/Few-shot Learning

Setting. The general setting of continual zero/few-shot text classification has been introduced in Sec. 2. Following a conventional continual learning setting (Lopez-Paz and Ranzato, 2017), a memory $\mathcal{M}_k$ is associated with a task $\mathcal{T}^{(k)}$ to store a fixed number of training examples[1] per seen task. Upon the arrival of a new task, given the label names in the task and optionally a support set, a generative model produces new task-specific examples. A classifier is trained on a combined set of examples from the support set, the memories, and the augmented examples, and evaluated on the test sets of all seen tasks. The datasets we used are EMPATHETIC and TACRED . Please refer to Appendix B.1 and B.2 for more details.

Evaluation. We use the widely adopted metric ACC_{avg} in continual learning, which measures the performance by averaging the accuracies of the classifier on test sets of all seen tasks $\{\mathcal{D}^{(1)}_{test},...,\mathcal{D}^{(k)}_{test}\}$, namely $\text{ACC}_{\text{avg}} = \frac{1}{k}\sum_{i=1}^{k} acc_i$ (Lopez-Paz and Ranzato, 2017). In addition, to measure the diversity of generated examples, we calculate the average similarity scores between all pairs of examples within each label, *i.e.* $\frac{1}{|\mathcal{Y}|}\sum_{i,j} sim(\boldsymbol{x}_i, \boldsymbol{x}_j)\mathbb{1}[\arg\max p(y|\boldsymbol{x}_i) == \arg\max p(y|\boldsymbol{x}_j)]$, where we use BLEU (Papineni et al., 2002) and word mover distance (WMD) (Kusner et al., 2015) as the similarity functions. The *lower* scores indicate more diversified examples within each label.

Baselines. We compare five data augmentation baselines: *i)* EDA (Wei and Zou, 2019) randomly deletes, substitutes, inserts or swaps words in the original sentences. *ii)* BERT (Ma, 2019) uses BERT to determine the position to insert or substitute words. *iii)* RTT (Sennrich et al., 2015) augments datasets by generating the paraphrases of the original sentences through round-trip translation, *iv)* LAMBDA (Kumar et al., 2020) trains a GPT2 to generate examples conditioned on the label text and uses a classifier to filter out low-quality examples as in our work. *v)* EX2 (Lee et al., 2021) applies T5 (Raffel et al., 2020) and extrapolation technique to increase the diversity of generated examples and deal with the low-resource setting. *vi)* OPTIMUS (Li et al., 2020) is our backbone model which is in an auto-encoder framework that uses BERT as the encoder and GPT2 (Radford et al., 2019) as the decoder. *vii)* CASUAL-LENS (Hu and Li, 2021) improves the training of OPTIMUS using an intervention and a counterfactual losses. Both OPTIMUS and CASUAL-LENS are designed for controllable text generation. We use them for data augmentation by assessing their ability for label-conditional generation.

[1]The examples in a memory are selected either from the corresponding support set or augmented data.

Methods	TACRED			EMPATHETIC		
	0	1	5	0	1	5
NO AUG	11.21	22.02	36.87	9.75	14.20	24.07
EDA	-	20.64	32.83	-	13.72	20.16
BERT	-	20.21	35.43	-	13.82	21.52
RTT	-	24.23	33.60	-	14.06	20.37
LAMBDA	16.23	20.16	32.14	13.39	14.45	21.95
EX2	15.57	19.83	32.87	11.91	16.82	24.75
OPTIMUS	17.12	19.99	28.77	9.93	14.32	18.21
CASUAL-LENS	9.83	17.17	25.76	9.72	11.72	16.49
VAE-DPRIOR	**31.34**	**37.31**	**44.17**	**18.08**	**22.71**	**31.82**

Table 1: The ACC_{avg} of the classifier across the tasks with different data augmentation methods. '-' indicates that zero-shot is not applicable to the corresponding augmentation methods.

Main Results and Discussions. We compare first the baselines with our model in its best setting, coined VAE-DPRIOR, which applies both the disentanglement priors and the MMD regularizer between the priors. The results in Table 1 show that it outperforms all data augmentation baselines on all zero/few-shot learning settings by significant margins. The augmentation approaches such as EDA , BERT and RTT generate adversarial examples of the original sentences via manipulation of words or paraphrasing. However, adversarial distributions are not the same as the true distribution, thus their generated examples do not improve the continual learning performance. They even degrade the performance in the five-shot setting in comparison to that without data augmentation.

Although LAMBDA , EX2 , OPTIMUS and CASUAL-LENS aim to learn the true distribution from labeled data, we observe that they often fail to generate texts in accordance with correct labels, especially for new tasks. Thus, their performance cannot be improved given more labeled examples of new tasks. In contrast, VAE-DPRIOR achieves a significantly higher degree of compositional generalization across tasks, evident by high average accuracy of the classifier trained on its generated examples. The performance of the classifier further

Datasets	Metrics	EDA	BERT	RTT	LAMBDA	EX2	OPTIMUS	CASUAL-LENS	VAE-DPRIOR
TACRED	BLEU↓	38.24	28.12	96.83	95.85	26.61	47.04	18.78	**4.14**
	WMD↓	97.91	96.83	99.93	99.28	**83.24**	94.94	88.24	86.08
EMPATHETIC	BLEU↓	31.64	24.97	96.79	74.82	44.55	55 12	9.46	**5.57**
	WMD↓	97.89	96.85	99.92	97.20	94.57	98.15	**79.10**	92.67

Table 2: The diversity scores of the generated examples measured with BLEU and WMD on one-shot learning.

improves when our model is fed with more labeled examples of new tasks.

We also evaluated *diversity* of the generated examples of all augmentation methods in the one-shot setting, presented in Table 2. The generated sentences from EDA , BERT and RTT are mostly paraphrases of the original sentences. Therefore, they cannot significantly diversify examples at the semantic level. LAMBDA generates data examples conditioning on the label names and the first k words of the original sentence, which also lacks diversity. EX2 enriches the diversity by extrapolating the novel samples from the existing sentences within the novel labels. OPTIMUS and CASUAL-LENS employ a GAN (Goodfellow et al., 2014) and a conditional GAN (Mirza and Osindero, 2014) respectively to generate diversified latent vectors for the generation of examples with the novel labels. However, with merely one or five sentences per label, such methods only generate a small sample of texts with novel labels. In contrast, VAE-DPRIOR can combine plenty of seen content representations acquired from the past with representations of new labels to generate high-quality sentences.

3.2 Ablation Study

Disentanglement. To show the importance of ϵ-disentanglement, we remove the constraint of the optimization problem (2) by using only the pre-trained model resulted from the first training step, denoted by VAE-DPRIOR (AE). As shown in Table 4, it suffers from a significant drop in terms of all metrics in the one-shot setting. In the same table, we also report the comparisons with alternative priors: (i) unconditional priors as in the vanilla VAE (VAE (UNCOND)), (ii) the same priors but increasing the variance coefficients of the two priors from 1 to 50 (VAE-DPRIOR (LGVAR)), (iii) a Gaussian mixture with randomly initialized means as the content prior but do not fine-tune the parameters of the prior (VAE-DPRIOR (RAND)), (iv) same as (iii) but fine-tune the parameters of the content prior (VAE-DPRIOR (RAND-FT)), and (v) a simple factorized Gaussian conditioned on the averaged sentence embedding of all sentences of initial tasks as the content prior (VAE-DPRIOR (GAUSS)). In

Methods	TACRED 0	TACRED 1	TACRED 5	EMPATHETIC 0	EMPATHETIC 1	EMPATHETIC 5
VAE-DPRIOR	31.34	37.31	44.17	18.08	22.71	31.82
- MMD	29.21	37.18	43.34	17.79	22.43	32.27
-/+ GAN	25.72	27.97	35.98	14.41	17.65	24.17
-/+ HSIC	8.46	14.09	42.25	18.02	21.39	32.28
-/+ IDEL	29.79	36.10	43.08	16.12	22.00	32.58

Table 3: The ACC_{avg} of VAE-DPRIOR with different disentanglement losses in zero/few-shot learning.

Methods	TACRED ACC_{avg} ↑	TACRED BLEU↓	TACRED WMD↓	EMPATHETIC ACC_{avg} ↑	EMPATHETIC BLEU↓	EMPATHETIC WMD↓
VAE-DPRIOR	37.31	4.14	86.08	22.71	5.57	92.67
-/+ VAE-DPRIOR (AE)	30.80	4.76	88.80	17.18	7.11	95.90
-/+ VAE-DPRIOR (GAUSS)	13.74	6.16	85.25	13.28	14.31	92.78
-/+ VAE-DPRIOR (LGVAR)	23.24	2.83	57.44	19.22	20.74	94.69
-/+ VAE-DPRIOR (RAND)	20.41	75.46	97.53	12.60	23.96	93.84
-/+ VAE-DPRIOR (RAND-FT)	23.79	87.31	98.87	15.30	88.60	99.31
-/+ VAE (UNCOND)	19.23	46.80	95.19	17.03	54.34	97.04
-/+ VQ-VAE	27.84	13.27	88.78	10.14	11.41	87.29
-/+ C-VAE	13.22	2.97	59.69	13.15	5.26	76.87
-/+ VAMP-VAE	19.08	30.47	82.31	17.47	41.00	95.75

Table 4: The ACC_{avg} and diversity scores of the models with different VAE frameworks on one-shot learning. Sample from prior label and content distributions.

the one-shot setting, the accuracy drops by more than 7% and 3% on TACRED and EMPATHETIC , respectively, using the alternative or no priors, indicating the importance of ϵ-disentanglement. Increasing the variance of our priors also jeopardize the ϵ-disentanglement. As evident in Fig. 3 using the t-SNE (Van der Maaten and Hinton, 2008), it is clear that the priors of VAE (UNCOND) and VAE-DPRIOR (LGVAR) are severely overlapped in contrast to VAE-DPRIOR.

We further investigate how the disentanglement regularizers influence our model by removing MMD or replacing MMD with GAN, HSIC, and IDEL (Cheng et al., 2020). As in Table 3, except for MMD, the other disentanglement regularizers bring almost no improvement to VAE-DPRIOR. HSIC, GAN and IDEL enforce independence between latent variables but even hurt the performance. We observe that the GAN-based regularizer causes mode collapse, because VAE-DPRIOR with GAN tends to generate overly similar examples. In contrast, if we apply the MMD to the other types of VAEs, such as a vanilla VAE, they lead to improved performance (see Appendix B.3) because the other VAEs do not have the ability to disentangle representations.

Posterior Collapse. Classical VAEs, such as vanilla VAE (VAE (UNCOND)), suffer from a notorious problem called *posterior collapse*. Those mod-

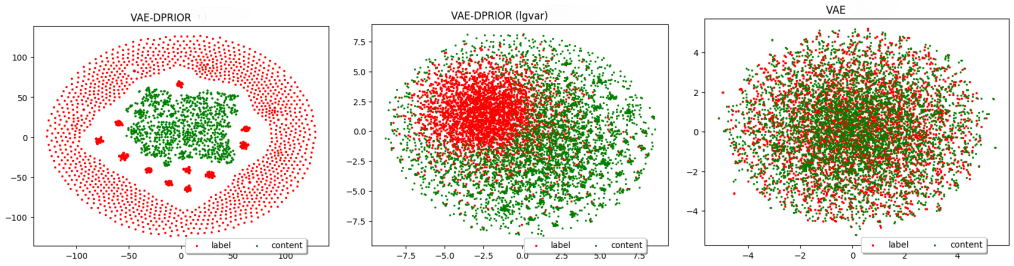

Figure 3: The label (red) and content (green) representations sampled from label and content priors of VAE-DPRIOR, VAE-DPRIOR (LGVAR) and VAE (UNCOND) trained on TACRED .

els learn non-injective mappings between latent variable values and the likelihoods; thus many latent representations are mapped to the same model outputs. We investigate this problem by comparing VAE-DPRIOR with VAE (UNCOND), VQ-VAE (Oord et al., 2017), C-VAE (Jang et al., 2016), VAMP-VAE (Tomczak and Welling, 2018) and VAE-DPRIOR with alternative priors described before in terms of diversity and accuracy. If there is severe posterior collapse, models will generate similar texts indicated by the high BLEU scores, and the classifier trained on the augmented data would perform poorly. Unsurprisingly, the results in Table 4 show that VAE-DPRIOR largely outperforms those VAEs. Although VAMP-VAE also introduces conditional priors, the latent variables of its priors do not require to be ϵ-disentangled with a small ϵ. The VAE-DPRIOR (RAND-FT) even generate almost identical texts.

Posterior collapse should lead to high ratios of duplicated outputs. Thus we feed each model with 200 diverse latent variable values randomly sampled from their priors and compute the duplicate ratios per label. VAE-DPRIOR (RAND-FT) has the highest ratio 97.38%, followed by VAE (UNCOND), VAMP-VAE , VAE-DPRIOR (RAND-FT), C-VAE and VQ-VAE with a duplicity ratio of 78.33%, 70.38%, 8.06%, 6.10% and 3.09%, respectively, on EMPATHETIC in the one-shot learning. In contrast, our model generates no duplicates with those latent variable values.

We also investigate the quality of the outputs of those models by sampling representations from the posterior distributions, $q_{\phi}(\mathbf{Z}_y|X, y)$ and $q_{\phi}(\mathbf{Z}_c|X, C)$. The duplicate ratio of VAE (UNCOND) drops to merely 4.27%, while that of VAMP-VAE increases to 97.95% on EMPATHETIC . Our model still achieves a zero duplicate ratio. However, appendix B.4 shows that the models sampling from posteriors achieve comparable results as those sampling from priors in terms of accuracy. Therefore, sampling contents from the posteriors may reduce duplicate ratios for some of the models but their generated examples still cannot have comparable quality as our model.

Methods	TACRED			EMPATHETIC		
	ACC_{avg} ↑	BLEU↓	WMD↓	ACC_{avg} ↑	BLEU↓	WMD↓
+ BERT + LABEL ENCODER	37.31	4.14	86.08	22.71	5.57	92.67
-/+ RANDOM	24.21	3.81	80.75	21.03	4.64	85.68
-/+ PERPLEXITY	23.32	4.72	80.98	16.59	4.81	73.94
-/+ BERT	33.21	3.96	81.81	20.40	7.81	75.77

Table 5: The ACC_{avg} and diversity scores of VAE-DPRIOR with different quality control methods on one-shot learning.

Quality Control. We compare our inference method (+BERT + LABEL ENCODER) with three alternative methods: *i)* only use a pre-trained BERT to encode each output text into an embedding with mean pooling and compare it with the average BERT embeddings of labels and support sets (+BERT); *ii)* select k examples with the lowest perplexity calculated by GPT2 (+PERPLEXITY); and *iii)* randomly select k outputs (+ RANDOM). Table 5 shows that different methods indeed influence the final performance of the data augmentation. Although the BLEU and WMD metrics show that baseline filtering methods all increase the diversity, the quality of selected examples is actually lower. But even with the worst performing filtering method, (+PERPLEXITY), our method can still outperform other baselines on both datasets. We observe that BERT with the encoder trained with label condition prior performs much better than only using the backbone model, (+BERT), in terms of selecting high quality examples, proving that the label prior condition could help the encoder generalize well on the novel labels.

4 Related Work

Variational Autoencoder. A large series of work learn representations based on generative models, such as Variational Autoencoder (VAE) (Kingma and Welling, 2019). A standard VAE minimizes the Kullback–Leibler divergence between the parametric posterior and the true posterior. Different

posterior integrates various properties to the generative models, VQ-VAE (Oord et al., 2017) parameterizes a posterior distribution of discrete latent random variables, Categorical-VAE (Jang et al., 2016) and Vamp-VAE (Tomczak and Welling, 2018) constructs mixture of posteriors on learnable pseudo-inputs. They are not capable of remembering rich contents in NLG. Gaussian Mixture VAE (Dilokthanakul et al., 2016) uses the mixture of Gaussian as the prior as well. It is not designed for disentanglement as it uses only one prior. An Identifiable Double VAE (Mita et al., 2021) does not use two different priors for different subspace. It uses only one prior based on observed random variables to remove the observed information from the latent representations in order to achieve disentanglement. In another word, it has no component to remember contents for continuous few shot setting. Our work considers the posterior on multiple types of disentangle random variables, which is potentially of more expressiveness.

Disentangled Representation Learning. Disentanglement of representations is one of the ultimate goals of deep learning. The existing methods are either unsupervised or supervised (Higgins et al., 2018). The unsupervised ones mainly fall into either the framework of VAE (Burgess et al., 2018) or Generative Adversarial Learing (GAN) (Tran et al., 2017). The recent works have also incorporate causality theories for robustness (Hu and Li, 2021). There are growing interests in applying disentangled representation learning in NLP applications, such as text style transfer (John et al., 2018a) and mitigating gender bias (Liu et al., 2020). However, it is challenging for those NLP approaches to work in the low-resource settings because they do not store rich content information inside models (Romanov et al., 2018).

Controllable Text Generation. Our method decomposes content and (attribute) label, where the label could be considered as additional control signal for text generation. We connect our work to those text style transfer (TST) and controllable text generation (CTG). *Representation disentanglement* is an important line of research in TST, which disentangles content and attribute representations (John et al., 2018a). Many disentanglement approaches are proposed to minimize the dependence between these two representations, such as mutual information (Yuan et al., 2020) and orthogonality (Wei et al., 2021). CTG controls the text generation of language models by smart prompt design (Li and Liang, 2021; Shin et al., 2020) or training conditioned on the controllable variables (Li et al., 2020; Hu and Li, 2021). Our work is highly aligned with (Li et al., 2020; Li and Liang, 2021). Since (Li et al., 2020) and (Li and Liang, 2021) have similar implementations, and (Li et al., 2020) was designed for generation conditioned on latent variables, we pick (Li et al., 2020) as one of our baselines.

5 Conclusion

In this work, we propose a VAE model with disentanglement priors for disentangled representation learning in low resource controllable NLG tasks. The disentanglement priors satisfy a novel property called ϵ-disentangled which builds a constraint space for the training problem. This model is able to effectively combine rich content representations sampled from a conditional content prior and task-specific representations for new tasks. Its empirical performance outperforms the baselines on continual zero-shot/few-shot text classification and few-shot text style transfer by a wide margin.

Limitations

We have studied ϵ-disentangled only in the VAE framework for task-specific language generation, though we believe it should be useful for a wide range of latent models. Although the content prior of our model can already be used to sample rich content representations, there is a possibility to store more information and represent a even richer content space reflected in real-world scenarios. In addition, our model has not considered application scenarios with limited computing resources. Though it is beyond the scope of this work, due to the heavy use of pre-trained large-scale language models, the deployment of our model in those cases is particularly challenging.

Acknowledgement

This material is based on research sponsored by Air Force Research Laboratory and DARPA under agreement numbers FA8750-19-2-0501 and HR001122C0029. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The computational resources of this work are supported by the Multi-modal Australian ScienceS Imaging and Visualisation Environment (MASSIVE).

References

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2020. Do not have enough data? deep learning to the rescue! In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7383–7390.

Antreas Antoniou, Massimiliano Patacchiola, Mateusz Ochal, and Amos Storkey. 2020. Defining benchmarks for continual few-shot learning. *arXiv preprint arXiv:2004.11967*.

Magdalena Biesialska, Katarzyna Biesialska, and Marta R. Costa-jussà. 2020. Continual lifelong learning in natural language processing: A survey. In *Proceedings of COLING*, pages 6523–6541.

Christopher M Bishop and Nasser M Nasrabadi. 2006. *Pattern recognition and machine learning*, volume 4. Springer.

Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. 2018. Understanding disentangling in beta-vae. *arXiv preprint arXiv:1804.03599*.

Pengyu Cheng, Martin Renqiang Min, Dinghan Shen, Christopher Malon, Yizhe Zhang, Yitong Li, and Lawrence Carin. 2020. Improving disentangled text representation learning with information-theoretic guidance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7530–7541.

Cyprien de Masson d'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. 2019. Episodic memory in lifelong language learning. In *Proceedings of NeurIPS*, pages 13122–13131.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Nat Dilokthanakul, Pedro AM Mediano, Marta Garnelo, Matthew CH Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. 2016. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*.

Songlin Dong, Xiaopeng Hong, Xiaoyu Tao, Xinyuan Chang, Xing Wei, and Yihong Gong. 2021. Few-shot class-incremental learning via relation knowledge distillation. In *Proceedings of AAAI*, pages 1255–1263.

Maziar Moradi Fard, Thibaut Thonet, and Eric Gaussier. 2020. Deep k-means: Jointly clustering with k-means and learning representations. *Pattern Recognition Letters*, 138:185–192.

Steven Y. Feng, Varun Gangal, Dongyeop Kang, Teruko Mitamura, and Eduard Hovy. 2020. GenAug: Data augmentation for finetuning text generators. In *Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 29–42.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard H. Hovy. 2021. A survey of data augmentation approaches for NLP. *CoRR*, abs/2105.03075.

Robert M French. 1999. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773.

Xu Han, Yi Dai, Tianyu Gao, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2020. Continual relation learning via episodic memory activation and reconsolidation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6429–6440.

Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. 2018. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*.

Zhiting Hu and Li Erran Li. 2021. A causal lens for controllable text generation. *Advances in Neural Information Processing Systems*, 34.

Yufan Huang, Yanzhe Zhang, Jiaao Chen, Xuezhi Wang, and Diyi Yang. 2021. Continual learning for text classification with information disentanglement based regularization. In *Proceedings of NAACL-HLT*, pages 2736–2746.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Di Jin, Zhijing Jin, Zhiting Hu, Olga Vechtomova, and Rada Mihalcea. 2020. Deep learning for text style transfer: A survey. *arXiv preprint arXiv:2011.00416*.

Xisen Jin, Mohammad Rostami, and Xiang Ren. 2021. Lifelong learning of few-shot learners across NLP tasks. *CoRR*, abs/2104.08808.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2018a. Disentangled representation learning for non-parallel text style transfer. *arXiv preprint arXiv:1808.04339*.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2018b. Disentangled representation learning for non-parallel text style transfer. *arXiv preprint arXiv:1808.04339*.

Zixuan Ke, Hu Xu, and Bing Liu. 2021. Adapting BERT for continual learning of a sequence of aspect sentiment classification tasks. In *Proceedings of NAACL-HLT*, pages 4746–4755.

Diederik P Kingma and Max Welling. 2019. An introduction to variational autoencoders. *arXiv preprint arXiv:1906.02691*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, pages 177–180.

Kalpesh Krishna, Deepak Nathani, Xavier Garcia, Bidisha Samanta, and Partha Talukdar. 2022. Few-shot controllable style transfer for low-resource multilingual settings. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7439–7468.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. *arXiv preprint arXiv:2003.02245*.

Varun Kumar, Hadrien Glaude, Cyprien de Lichy, and Wlliam Campbell. 2019. A closer look at feature space data augmentation for few-shot intent classification. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 1–10.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International conference on machine learning*, pages 957–966. PMLR.

Kenton Lee, Kelvin Guu, Luheng He, Tim Dozat, and Hyung Won Chung. 2021. Neural data augmentation via example extrapolation. *arXiv preprint arXiv:2102.01335*.

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

Han Li, Jihwan Lee, Sidharth Mudgal, Ruhi Sarikaya, and Young-Bum Kim. 2019. Continuous learning for large-scale personalized domain classification. In *Proceedings of NAACL-HLT*, pages 3784–3794.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Zhuang Li, Lizhen Qu, and Gholamreza Haffari. 2021. Total recall: a customized continual learning method for neural semantic parsers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3816–3831.

Zhaojiang Lin, Andrea Madotto, and Pascale Fung. 2020. Exploring versatile generative language model via parameter-efficient transfer learning. In *EMNLP Findings*, pages 441–459.

Haochen Liu, Wentao Wang, Yiqi Wang, Hui Liu, Zitao Liu, and Jiliang Tang. 2020. Mitigating gender bias for neural dialogue generation with adversarial learning. *arXiv preprint arXiv:2009.13028*.

David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30:6467–6476.

Edward Ma. 2019. Nlp augmentation. https://github.com/makcedward/nlpaug.

Remi Mir, Bjarke Felbo, Nick Obradovich, and Iyad Rahwan. 2019. Evaluating style transfer for text. *arXiv preprint arXiv:1904.02295*.

Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Graziano Mita, Maurizio Filippone, and Pietro Michiardi. 2021. An identifiable double vae for disentangled representations. In *International Conference on Machine Learning*, pages 7769–7779. PMLR.

Sharmila Reddy Nangi, Niyati Chhaya, Sopan Khosla, Nikhil Kaushik, and Harshit Nyati. 2021. Counterfactuals to control latent disentangled text representations for style transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 40–48.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2017. Neural discrete representation learning. *arXiv preprint arXiv:1711.00937*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Alexey Romanov, Anna Rumshisky, Anna Rogers, and David Donahue. 2018. Adversarial decomposition of text representation. *arXiv preprint arXiv:1808.09042*.

Halsey Lawrence Royden and Patrick Fitzpatrick. 1988. *Real analysis*, volume 32. Macmillan New York.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*.

Fatemeh Shiri, Terry Yue Zhuo, Zhuang Li, Shirui Pan, Weiqing Wang, Reza Haffari, Yuan-Fang Li, and Van Nguyen. 2022. Paraphrasing techniques for maritime qa system. In *2022 25th International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE.

Eric Michael Smith, Diana Gonzalez-Rico, Emily Dinan, and Y-Lan Boureau. 2019. Zero-shot fine-grained style transfer: Leveraging distributed continuous style representations to transfer to unseen styles. *arXiv preprint arXiv:1911.03914*.

Jakub Tomczak and Max Welling. 2018. Vae with a vampprior. In *International Conference on Artificial Intelligence and Statistics*, pages 1214–1223. PMLR.

Luan Tran, Xi Yin, and Xiaoming Liu. 2017. Disentangled representation learning gan for pose-invariant face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1415–1424.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962*.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Hong Wang, Wenhan Xiong, Mo Yu, Xiaoxiao Guo, Shiyu Chang, and William Yang Wang. 2019. Sentence embedding alignment for lifelong relation extraction. *arXiv preprint arXiv:1903.02588*.

Xinyi Wang and Yi Yang. 2020. Neural topic model with attention for supervised learning. In *International Conference on Artificial Intelligence and Statistics*, pages 1147–1156. PMLR.

Yixin Wang, David Blei, and John Patrick Cunningham. 2020. Posterior collapse and latent variable non-identifiability. In *Third Symposium on Advances in Approximate Bayesian Inference*.

Yixin Wang and Michael I Jordan. 2021. Desiderata for representation learning: A causal perspective. *arXiv preprint arXiv:2109.03795*.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*.

Yuxiang Wei, Yupeng Shi, Xiao Liu, Zhilong Ji, Yuan Gao, Zhongqin Wu, and Wangmeng Zuo. 2021. Orthogonal jacobian regularization for unsupervised disentanglement in image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6721–6730.

Tongtong Wu, Massimo Caccia, Zhuang Li, Yuan-Fang Li, Guilin Qi, and Gholamreza Haffari. 2022. Pre-trained language model in continual learning: A comparative study. In *International Conference on Learning Representations*.

Tongtong Wu, Xuekai Li, Yuan-Fang Li, Reza Haffari, Guilin Qi, Yujin Zhu, and Guoqiang Xu. 2021. Curriculum-meta learning for order-robust continual relation extraction. *CoRR*, abs/2101.01926.

Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Conditional bert contextual augmentation. In *International Conference on Computational Science*, pages 84–95. Springer.

Mengzhou Xia, Xiang Kong, Antonios Anastasopoulos, and Graham Neubig. 2019. Generalized data augmentation for low-resource translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5786–5796.

Pauching Yap, Hippolyt Ritter, and David Barber. 2021. Addressing catastrophic forgetting in few-shot problems. In *Proceedings of ICML*, pages 11909–11919.

Sung Whan Yoon, Do-Yeon Kim, Jun Seo, and Jaekyun Moon. 2020. XtarNet: Learning to extract task-adaptive representation for incremental few-shot learning. In *Proceedings of ICML*, pages 10852–10860.

Siyang Yuan, Pengyu Cheng, Ruiyi Zhang, Weituo Hao, Zhe Gan, and Lawrence Carin. 2020. Improving zero-shot voice style transfer via disentangled representation learning. In *International Conference on Learning Representations*.

Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. 2022. A survey of controllable text generation using transformer-based pre-trained language models. *arXiv preprint arXiv:2201.05337*.

Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

A Implementation Details

We use learning rate of 5e-5 for our method. The training epochs for our generation model in continual few-shot learning are 120 and 160 for EMPATHETIC and TACRED , respectively. All experiments are run five times with different random seeds and we report the average accuracies. The number of clusters for the deep content clustering loss are 1600, 800 and 3200 when training model on EMPATHETIC , TACRED and PERSONALITY , respectively. All the methods are trained on the V100 GPUs. The number of total parameters is 455864068 and the total number of total trainable parameters is 401751808. For style transfer, the training epochs are 120 and 160 for EMPATHETIC and PERSONALITY , respectively. We use BERT-small (Turc et al., 2019) as the backbone of the label and content encoders and GPT2-medium as the decoder. For data augmentation in continual few-shot learning, each label is augmented with 50 examples generated by different augmentation methods. OPTIMUS is not designed for style transfer. We adapt it to conduct style transfer by prepending a label phrase as a prompt before the input sentence. Style transfer can be done by altering the current label phrase to novel labels in the new tasks.

B Continual Few-shot Learning

B.1 Setting

We consider a continual few-shot learning setting similar as (Antoniou et al., 2020). The text classification model $\pi_\theta^c : \mathcal{X} \rightarrow \mathcal{Y}$ is trained sequentially on K distinct tasks $\{\mathcal{T}^{(1)}, \mathcal{T}^{(2)},...,\mathcal{T}^{(K)}\}$. The initial task $\mathcal{T}^{(1)}$ includes the training and test set $(\mathcal{D}^{(1)}_{train}, \mathcal{D}^{(1)}_{test})$ while the succeed tasks $\mathcal{T}^{(k>1)}$ includes the support and test sets $(\mathcal{D}^{(k)}_{sup}, \mathcal{D}^{(k)}_{test})$, where we assume $\mathcal{D}^{(1)}_{train}$ includes enough training data for each base class while $\mathcal{D}^{(k)}_{sup} = \{\boldsymbol{x}, \boldsymbol{y}\}_{i=1}^{N\times|C_k|}$ includes only N-shot instances per new class. The classes on $\mathcal{T}^{(k)}$ are disjoint from classes of previous tasks, $C_{1:k-1} \cap C_k = \emptyset$. As a conventional continual learning setting as in (Lopez-Paz and Ranzato, 2017), a memory $\mathcal{M}_k$ is associated with $\mathcal{T}^{(k)}$ to store a fixed number of training instances (either examples selected from the support sets or the synthetic data) per each seen task.

Upon arrival of each task, the classifier π_θ^c is trained on a combined set of instances from the support set, memory set, and the augmented examples generated by our generative model. To generate augmented examples, we sample content from the fixed clustering obtained using the large training data in $\mathcal{T}^{(1)}$, and sample labels from $C_{1:k}$. Note that for our model, as long as we have the generative model and store the label embeddings, we could regenerate examples from all old tasks. Therefore, the memory is not necessary for our model. But for a fair comparison with other baselines, we still assume there is fixed memory for each old task and use only the examples from this memory to replay the classifier. We apply our data augmentation method to EMAR (Han et al., 2020), a SOTA continual learning approach for text classification. We follow EMAR (Han et al., 2020) for the classifier architecture and how to update its parameters in continual learning.

B.2 Datasets

TACRED is a relation detection dataset which includes 42 relations. Following the settings in (Wang et al., 2019; Han et al., 2020), examples are clustered into ten groups given the word embeddings of the label phrases. 5 groups are randomly selected as the initial task. The support and test set from each task are drawn from each of the rest tasks. We randomly generate the support sets five times with different random seeds as well. Each support set includes 0, 1, or 5 examples.

B.3 Influence of MMD on VAEs

Table 6 shows the performance of VAEs on five-shot learning of TACRED with or without MMD. We present only five-shot results as we found that the MMD brings almost no improvement to different VAEs on zero/one-shot learning. But it consistently leads to performance improvement on all VAEs except for VAE-DPRIOR when the number of shots increases. We conjecture that disentanglement regularization perform better when there is enough label-specific information.

ACC_{avg}	VAE-DPRIOR	VAE (UNCOND)	VQ-VAE	C-VAE	VAMP-VAE
w/o MMD	43.35	27.86	35.27	30.77	34.38
w/ MMD	43.13	29.72	36.27	33.32	35.91

Table 6: The ACC_{avg} of VAE-DPRIOR and other VAEs with or without MMD of five-shot learning on TACRED . The representations are sampled from the label and latent representations from the posterior distributions.

Methods	TACRED			EMPATHETIC		
	ACC_{avg} ↑	BLEU↓	WMD↓	ACC_{avg} ↑	BLEU↓	WMD↓
VAE-DPRIOR	37.11	4.06	86.64	28.40	5.10	88.74
-/+ VAE-DPRIOR (GAUSS)	20.21	57.73	96.83	15.67	64.27	98.10
-/+ VAE-DPRIOR (RAND)	26.13	68.69	97.02	14.20	100.00	100.00
-/+ VAE-DPRIOR (RAND-FT)	25.82	40.49	94.16	17.31	50.49	96.93
-/+ VAE (UNCOND)	27.06	11.78	91.30	18.60	20.74	94.69
-/+ VQ-VAE	28.12	12.68	88.28	11.85	10.03	86.27
-/+ C-VAE	16.82	5.49	69.28	13.80	10.31	89.51
-/+ VAMP-VAE	17.46	100	100	16.58	98.19	99.92

Table 7: The ACC_{avg} and diversity scores of the models with different VAE frameworks on one-shot learning. The representations are sampled from the posterior label and content distributions.

B.4 Influence of Sampling from Posterior Distribution

Table 7 shows the performance of the classifiers using augmented data sampled from posterior distributions of VAEs. Since sampling representations from posterior distributions, $q_{\phi}(\mathbf{Z}_y|X, y)$ and $q_{\phi}(\mathbf{Z}_c|X, C)$, requires text X as the input, we feed all the text from the training sets in previous tasks to the content encoder to obtain the content representations and the text in the support set of new tasks to the label encoder to obtain the label representations. We combine the two types of representations to get augmented data for new tasks. Notice that the VAE-DPRIOR (AE) setting in Table 4 adopts a similiar way to sample examples for data augmentation except that the content and label representations $\boldsymbol{z}_c$ and $\boldsymbol{z}_y$ are generated by using **LabelEncoder** and **ContentEncoder** directly.

B.5 Accuracies of VAE-DPRIOR on PERSONALITY and FEWREL

Table 8 shows the performance of the baselines and VAE-DPRIOR on one-shot learning of PERSONALITY and FEWREL . Please notice that we use the exact same FEWREL dataset as in (Wang et al., 2019; Han et al., 2020) except that we split the tasks in a different way. VAE-DPRIOR performs best on both datasets as well.

ACC_{avg}	NO AUG	EDA	BERT	RTT	LAMBDA	EX2	OPTIMUS	CASUAL-LENS	VAE-DPRIOR
PERSONALITY	8.65	10.22	10.52	11.40	10.58	12.51	9.15	7.74	15.48
FEWREL	42.64	46.88	53.93	46.45	46.71	45.09	34.08	19.82	71.81

Table 8: The ACC_{avg} of the baselines and VAE-DPRIOR in one-shot learning on PERSONALITY and FEWREL .

B.6 Accuracies of VAE-DPRIOR using Soft and Hard EM

Table. 9 shows the VAE-DPRIOR on two datasets with soft and hard EM. The temperature τ for soft EM is set as 0.5. VAE-DPRIOR with soft EM has higher performance. However, we select the hard EM as our main setting because it brings a faster training speed.

C Few-shot Text-style Transfer

C.1 Setting

We follow the common non-parallel text style transfer setting as in (Nangi et al., 2021), where each text sample $\boldsymbol{x}$ is associated with a style label $\boldsymbol{y}$. In the few-shot setting, the style transfer model $\pi^s_\theta : \mathcal{X} \rightarrow \mathcal{X}'$ is pre-trained on a training set, $\mathcal{D}_{train}$, which includes abundant training data (*e.g.* more than 50 instances

ACC_{avg}	TACRED	EMPATHETIC
Hard EM	37.31	22.71
Soft EM	39.47	25.43

Table 9: The ACC_{avg} of VAE-DPRIOR in one-shot learning on TACRED and EMPATHETIC with hard and soft EM.

per style) for each base style C_b. After pre-training, the model parameters would be frozen with only the label embeddings updated based on a support set, $\mathcal{D}_{sup} = \{\boldsymbol{x}, \boldsymbol{y}\}_{i=1}^{N \times |C_n|}$, which includes only N-shot instances for each one of $|C_n|$ novel styles. Although VAE-DPRIOR can be easily fine-tuned on the support sets, we found the fine-tuning brings negligible performance gain in the few-shot setting. The test set $\mathcal{D}_{test}$ is sampled from both $\mathcal{D}_{train}$ and a corpus $\mathcal{D}'_{train}$ which is from the same distribution of $\mathcal{D}_{train}$. The style transfer task is to transfer text in $\mathcal{D}_{test}$ into the styles of C_n in $\mathcal{D}_{sup}$.

C.2 Datasets

EMPATHETIC dataset includes around 18,000 dialogues. Each dialogue consists of a context description and an associated empathetic type. PERSONALITY includes 200,000 image captions associated with 215 personality types. Since many personalities are highly correlated in terms of their semantics, we cluster these personalities into 35 groups and manually select one type for each group. For EMPATHETIC , we use the context descriptions as the original text to be transferred and their corresponding empathetic types as the styles. For PERSONALITY , we use the image captions and their personalities. We randomly select examples of 28 empathetic types and 30 personality types in the training set and draw support sets from the rest empathetic and personalities types. We draw 0, 1 or 5 examples for each held out label. After drawing, the rest examples are considered as the test examples. For each k-shot, the support and test sets are drawn five times with different random seeds to avoid bias during evaluation. Our experiments would be run on all the support sets and obtain the average performance.

C.3 Evaluation.

We use three automatic metrics, *Style-transfer Accuracy*, *Self-WMD* and *Perplexity*, to evaluate the accuracy of style transfer, semantic relevance and naturalness of the generated text, respectively. For *Style-transfer Accuracy*, we train a BERT (Devlin et al., 2018) classifier on styles. The averaged accuracy on target labels indicates the correctness of style transfer. *Self-WMD* (Kusner et al., 2015) measures WMD between the original text and the transferred text. *Perplexity* is estimated by a statistical language model in English released by (Koehn et al., 2007) [2].

C.4 Baselines.

We compare five style transfer baselines: *i)* R-VAE-AVG (John et al., 2018b) learns the disentangled label and content representations. *ii)* R-VAE-CF (Nangi et al., 2021), on the base of R-VAE-AVG , uses a counterfactual reasoning module to control the generation of label representations. *iii)* ZF (Smith et al., 2019) is a back-translation model, which aims to deal with the zero-shot text transfer problem. The two controllable text generation baselines used in the continual few-shot setting, *iv)* OPTIMUS and *v)* CASUAL-LENS , are extended for style transfer as well. Please refer to the Appendix A and their original work for the detailed style transfer implementation.

C.5 Inference.

The inference of VAE-DPRIOR for the text style transfer differs from the inference for continual few-shot learning. Given a new style, we start with sampling a name representation and text representations from the posterior label distribution, $q_{\phi}(\mathbf{Z}_y|X, y)$, conditioned on its associated name phrase and text sequences in the support set, respectively. Then, we create the label representation $\boldsymbol{z}_y$ for the new style by averaging its associated text representations and its name representation. The content representations are sampled from the posterior content distribution, $q_{\phi}(\mathbf{Z}_c|X, C)$, conditioned on the text to be style-transferred. After

[2] https://www.statmt.org/moses/RELEASE-4.0/models/cs-en/lm/

Methods	EMPATHETIC			PERSONALITY		
	Style Accuracy↑	Self-WMD↑	Perplexity↓	Style Accuracy↑	Self-WMD↑	Perplexity↓
R-VAE-AVG	28.49	90.19	627.49	27.73	86.06	796.99
R-VAE-CF	26.85	90.82	657.22	19.69	87.25	850.86
ZF	25.58	90.05	943.56	20.92	83.37	765.58
OPTIMUS	27.54	94.03	829.43	18.25	**94.96**	919.58
CASUAL-LENS	**39.27**	88.74	1157.28	20.52	89.62	1059.17
VAE-DPRIOR	32.67	**95.69**	**236.20**	**49.58**	93.44	**311.07**

Table 10: The results of one-shot style-transfer on both datasets.

feeding the content representations and the representations of target styles to the generator, we obtain the most likely outputs by beam-search.

C.6 Main Results and Discussions.

The results in Table 10 show that our method performs better than all baselines in terms of all metrics except *Style Accuracy* on EMPATHETIC and *Self-WMD* on PERSONALITY . An ideal style transfer model should find a good balance in terms of all three evaluation metrics. Though CASUAL-LENS and OPTIMUS can achieve the best on a single metric, they fail to perform well across all the metrics. We inspect that CASUAL-LENS performs poorly on preserving the content of the original sentence while OPTIMUS performs poor on style transfer and basically replicates the original sentences in PERSONALITY dataset. In contrast, the average ranking of VAE-DPRIOR on three metrics are highest among all baselines. Our model performs particularly well in terms of semantic relevance and naturalness while still keeping high accuracies of style transfer. Other methods that utilize disentanglement learning, including R-VAE-AVG , R-VAE-CF and CASUAL-LENS , often perform well on one metric while lose on the other metrics. We conjecture this is due to their methods do not fully disentangle the representations so they can not balance well between content preservation and style transfer.

C.7 Complete Automatic Evaluation Results of Style Transfer on two Datasets

The full results of automatic evaluation on Empathetic dataset and Personality dataset are presented in Table 11 and Tabel 12 respectively. Overall, on all few-shot settings, our method perform the best in terms of the average rank among all baselines. Although on EMPATHETIC dataset, R-VAE-AVG and CASUAL-LENS outperform our method in term of the Style Accuracy. Through inspection, we found that R-VAE-AVG and CASUAL-LENS tend to overfit to support set after finetuning merely on a small number of training instances. For example, R-VAE-AVG tends to copy the text from support set, which gains higher Style Transfer Accuracy. But this effect makes the Perplexity and Self-WMD of R-VAE-AVG and CASUAL-LENS decreasing from zero-shot to five-shot learning. In contrast, VAE-DPRIOR performs steady across different(zero/few-shot) settings. The Style Accuracy of VAE-DPRIOR is increasing without losing performance on content preservation and naturalness of sentences.

Methods	Style Accuracy↑			Self-WMD↑			Perplexity↓			AVG Rank↓		
	0	1	5	0	1	5	0	1	5	0	1	5
R-VAE-AVG	33.99	28.49	41.90	91.39	90.19	90.03	796.47	627.49	896.19	3.3	3.00	3.33
R-VAE-CF	20.39	26.85	26.99	93.95	90.82	90.75	825.80	657.22	858.61	4.67	3.67	4
ZF	21.85	25.58	26.60	93.64	90.05	89.51	785.29	943.56	609.68	3.67	5.33	4.33
OPTIMUS	26.26	27.54	27.50	94.24	94.03	93.91	814.45	829.43	826.63	3.33	3.33	3
CASUAL-LENS	34.53	39.27	41.32	89.74	88.74	88.42	1236.31	1157.28	1332.15	4.33	4.33	4.67
VAE-DPRIOR	32.84	32.67	34.55	95.68	95.69	95.70	233.03	236.20	233.66	1.67	1.33	1.67

Table 11: The results of zero, one and five-shot learning of style transfer on EMPATHETIC dataset.

C.8 Human evaluation result

We hire three crowd-workers to rate the sentences with score from 1-5 to indicate whether the the generated sentences belong to the target styles and whether the content of generated sentences are consistent with the original sentences. To evaluate naturalness, we follow the evaluation setting in (Mir et al., 2019) to let the crowd-workers distinguish the human generated sentences from the model generated sentences.

Methods	Style Accuracy↑			Self-WMD↑			Perplexity↓			AVG Rank↓		
	0	1	5	0	1	5	0	1	5	0	1	5
R-VAE-AVG	21.84	27.73	21.20	91.94	86.06	86.91	832.40	796.99	1202.90	3	3.33	4.33
R-VAE-CF	19.88	19.69	18.72	95.62	87.25	87.92	852.62	850.86	1205.90	2.67	4.33	5
ZF	16.98	20.92	21.80	87.45	83.37	83.63	643.00	765.58	891.42	4.67	3.67	3.67
OPTIMUS	17.79	18.25	19.21	95.32	94.96	95.00	966.29	919.58	955.78	4	4	3
CASUAL-LENS	19.46	20.52	22.69	91.30	89.62	89.15	1238.83	1059.17	1567.96	5.00	4.33	3.67
VAE-DPRIOR	50.64	49.58	55.51	93.69	93.44	93.46	322.03	311.07	304.36	1.67	1.33	1.33

Table 12: The results of zero, one and five-shot learning of style transfer on PERSONALITY dataset.

Method	Content↑	Style↑	Nature↓	Rank↓
R-VAE-CF	1.33	2.26	0.45	2.67
R-VAE-AVG	1.20	2.15	**0.44**	3.33
ZF	1.17	2.01	0.57	5.33
OPTIMUS	2.33	1.85	0.56	4
CASUAL-LENS	2.04	2.24	0.58	4
VAE-DPRIOR	**2.44**	**2.40**	0.52	**1.67**

Table 13: Human evaluation results, evaluated by content preservation (Content), style transfer correctness (Style), naturalness (Nature), and average rank of the three criterions (Rank).

The naturalness score in Tab. 13 indicates successful rate of distinguishing the sentences. The easier the sentence is distinguished, the less natural the sentence is. We achieve far superior performance in terms of both Content Preservation and Style Transfer metrics. Although on Naturalness, our method only ranks third. We conjecture that the generated sentences by VAE-DPRIOR is usually longer than the original sentences. The crowd-workers could easily grasp this pattern and distinguish the sentences. Besides, with Naturalness metric, the gap between different methods are actually insignificant, which are all close to 50%.

C.9 Generated Examples of Style Transfer

Methods	Original Style: prepared → Target Style: anticipating
ZF	i have m so scared of spiders. i can't stand those things!
R-VAE-CF	i was schocked to see my favorite band wasnt coming to my city this tour
R-VAE-AVG	i cannot wait until next month. i had a feeling my birthday was.
OPTIMUS	I thought I didn't planned for my job interview at jobster trip . I felt like going off
CASUAL-LENS	I felt very apprehensive when I went to my interview
VAE-DPRIOR	I felt really good at my job interview at work today I felt I did well at the job I worked out for when I saw I had done well in my interview for the position I was looking forward to doing at that time .

Table 14: The style transfer results of different models trained on dataset EMPATHETIC with one-shot learning setting. The original sentence, "i felt like i did well at my job interview yesterday. i went in feeling confident", is transferred from the original style "prepared" to the target style "anticipating".

Table 14 and 15 depict the examples of generate examples of different style transfer methods trained on EMPATHETIC and PERSONALITY , respectively.

D Related Work Supplementary

Data Augmentation. Data augmentation (DA) encompasses methods of increasing training data diversity without directly collecting more data (Feng et al., 2021), which can be roughly categorized into (1) rule-based methods (Wei and Zou, 2019), (2) example interpolation methods (Zhang et al., 2018), and

Methods	Original Style: appreciative → Target Style: angry
R-VAE-CF	the amount of shadows in the middle of.
R-VAE-AVG	the amount of players in the left of building.
ZF	these mountains just look so nice! i would love to see them.
OPTIMUS	fl in the fissip , the fissile columns in theTyphris ' windows in the Sky .
CASUAL-LENS	the horizon is filled with dazzling colors .
VAE-DPRIOR	Look at the splashes on the rocks in the middle of the street , I hate looking at rocks . They're so ugly looking , and I can't stand to look at them anymore .

Table 15: The style transfer results of different models trained on dataset PERSONALITY with one-shot learning setting. The original sentence, "Look at the fissures in the strata columns, beautiful.", is transferred from the original style "appreciative" to the target style "angry".

(3) model-based methods (Wu et al., 2019; Sennrich et al., 2015; Anaby-Tavor et al., 2020; Kumar et al., 2020; Lee et al., 2021; Shiri et al., 2022). Data augmentation generally encourages better performance in low-resource scenarios, such as few-shot learning (Kumar et al., 2019) and low-resource language learning (Xia et al., 2019). Although data augmentation has been well applied in many tasks (Feng et al., 2021), there has been limited work on DA for conditional text generation (Feng et al., 2020).

Continual Few-shot Learning. The primary challenge addressed in continual learning literature is overcoming catastrophic forgetting (French, 1999; Biesialska et al., 2020; Wu et al., 2022), Various approaches have been proposed to tackle the forgetting problem, e.g., rehearsal-based methods (Han et al., 2020; de Masson d'Autume et al., 2019; Li et al., 2021; Wu et al., 2021), regularization-based methods (Li et al., 2019; Huang et al., 2021), and dynamic architecture methods (Ke et al., 2021; Lin et al., 2020). Continual few-shot learning is an even more challenging yet realistic setting which encourages learners the quick adaptation ability during learning (Jin et al., 2021; Yoon et al., 2020). Comparing to the numerous researches out of NLP applications (Yap et al., 2021; Yoon et al., 2020; Dong et al., 2021), continual few-shot language learning is still an under-explored area (Jin et al., 2021).

E Proofs

E.1 Discussion about ϵ-Disentangled

To achieve information purity, the learned models should follow the structure illustrated in Fig.2(a) that there is no dependency between C and $\mathbf{Z}_y$, and similarly no dependency between y and $\mathbf{Z}_c$. However, prior works on disentangled representation learning regularize the models by minimizing mutual information $I(\mathbf{Z}_c, \mathbf{Z}_y)$ between $\mathbf{Z}_c$ and $\mathbf{Z}_y$ such that $\mathbf{Z}_c \perp\!\!\!\perp \mathbf{Z}_y$ when $I(\mathbf{Z}_c, \mathbf{Z}_y) = 0$ (Cheng et al., 2020; Wang and Jordan, 2021). In another word, prior works only require that there is no edge between $\mathbf{Z}_c$ and $\mathbf{Z}_y$ in the Bayesian model. However, this does not imply $I(\mathbf{Z}_c, y) = 0$ and $I(\mathbf{Z}_y, C) = 0$. In the trained models, C can still be the shared parent or child of two independent random variables using the regularization from the prior works. In addition, the independence assumption between $\mathbf{Z}_c$ and $\mathbf{Z}_y$ does not always hold in practice. For example, if $\mathbf{Z}_y$ is a random variable for emotion categories and $\mathbf{Z}_c$ represents events influencing emotions, they are causally dependent. Forcing the independent assumption may deteriorate model performance.

To address this limitation, we propose to regularize the priors of latent variables for encouraging information purity. If we have a close look at $I(\mathbf{Z}_y, y) = \int p(\mathbf{Z}_y, y) \log \frac{p(\mathbf{Z}_y, y)}{p(\mathbf{Z}_y)p(y)}$, which is simplified to $\int p(y|\mathbf{Z}_y)p(\mathbf{Z}_y) \log \frac{p(\mathbf{Z}_y|y)}{p(\mathbf{Z}_y)}$, a high mutual information expects $p(\mathbf{Z}_y) > 0$ whenever $p(\mathbf{Z}_y|y)$ is high. Similarly, if we aim for an extremely small $I(\mathbf{Z}_y, C)$, we expect a low $p(\mathbf{Z}_y)$ or $p(\mathbf{Z}_y) = 0$ whenever $p(\mathbf{Z}_y|C) > p(\mathbf{Z}_y)$. If we design the priors in the way that their dense regions are not overlapped, we achieve information purity by maximizing the corresponding mutual information.

We do not require **absolute continuity** for the associated divergence measure because when the priors are ϵ-disentangled with a fairly low ϵ, one of the priors would have zero probability in the regions where

the other prior has positive supports.

E.2 Latent Variable Non-identifiability

Wang et al. (2020) introduce the concept of latent variable non-identifiability and shows that it leads to posterior collapse.

Definition E.1 (Latent variable non-identifiability (Wang et al., 2020))**.** Given a likelihood function $p_{\boldsymbol{\theta}}(\mathbf{X}|\mathbf{Z};\boldsymbol{\theta})$ with parameters $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$ and a dataset $\mathcal{D} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$, the latent variables $\mathbf{Z}$ are non-identifiable if $p(\mathcal{D}|\mathbf{Z} = \boldsymbol{z}; \hat{\boldsymbol{\theta}}) = p(\mathcal{D}|\mathbf{Z} = \boldsymbol{z}'; \hat{\boldsymbol{\theta}})\ \forall \boldsymbol{z}, \boldsymbol{z}' \in \mathcal{Z}$, where $\mathcal{Z}$ denotes the space of latent variable values. As a consequence, $p(\mathcal{D}|\mathbf{Z}; \hat{\boldsymbol{\theta}}) = p(\mathcal{D}; \hat{\boldsymbol{\theta}})$.

For the cases with more than one random variables (vectors), we extend this idea for latent condtional non-identifiability.

Definition E.2 (Latent variable conditional non-identifiability)**.** Given a likelihood function $p_{\boldsymbol{\theta}}(\mathbf{X}|\mathbf{Z}_a, \mathbf{Z}_b;\boldsymbol{\theta})$ with parameters $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$ and a dataset $\mathcal{D} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$, the latent variables $\mathbf{Z}_a$ are non-identifiable conditioned on $\mathbf{Z}_b$ if $p(\mathcal{D}|\mathbf{Z}_a, \mathbf{Z}_b; \hat{\boldsymbol{\theta}}) = p(\mathcal{D}|\mathbf{Z}_b; \hat{\boldsymbol{\theta}})$.

Proposition E.3. *Given a likelihood function $p_{\boldsymbol{\theta}}(\mathbf{X}|\mathbf{Z}_a, \mathbf{Z}_b;\boldsymbol{\theta})$ with parameters $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$ and a dataset $\mathcal{D} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$, the latent variables $\mathbf{Z}_a$ are non-identifiable conditioned on $\mathbf{Z}_b$ if $p(\mathbf{Z}_a|\mathbf{Z}_b; \hat{\boldsymbol{\theta}}) = 1$.*

Proof:
Given $p(\mathbf{Z}_a|\mathbf{Z}_b; \hat{\boldsymbol{\theta}}) = 1$,

$$p(\mathcal{D}|\mathbf{Z}_b; \hat{\boldsymbol{\theta}})p(\mathbf{Z}_a|\mathbf{Z}_b; \hat{\boldsymbol{\theta}}) = p(\mathcal{D}|\mathbf{Z}_a, \mathbf{Z}_b; \hat{\boldsymbol{\theta}})$$
$$p(\mathcal{D}|\mathbf{Z}_b; \hat{\boldsymbol{\theta}}) = p(\mathcal{D}|\mathbf{Z}_a, \mathbf{Z}_b; \hat{\boldsymbol{\theta}})$$

For all $\boldsymbol{x}_i$ in $\mathcal{D}$, if $p(\boldsymbol{x}_i|\mathbf{Z}_a = \boldsymbol{z}, \mathbf{Z}_b = \boldsymbol{z}) > p(\boldsymbol{x}_i|\mathbf{Z}_a = \boldsymbol{z}, \mathbf{Z}_b = \boldsymbol{z}')$ with $\boldsymbol{z} \neq \boldsymbol{z}'$ for all $\boldsymbol{z}, \boldsymbol{z}' \in \mathcal{Z}$, where $\mathcal{Z}$ is the space of latent variable values, $p(\mathbf{Z}_a|\mathbf{Z}_b)$ is high because both random variable vectors are almost a copy to each other. In this case, if $p_a(\mathbf{Z}_a)$ and $p_b(\mathbf{Z}_b)$ share the same dense regions or even the same, such a conditional non-identifiable case will not be penalized during training. In contrast, if $p_a(\mathbf{Z}_a)$ and $p_b(\mathbf{Z}_b)$ are ϵ-disentangled with a small ϵ, the parameters leading to the conditional non-identifiable cases are disencouraged by receiving zero or a low likelihood $p(\boldsymbol{x}_i|\mathbf{Z}_a = \boldsymbol{z}, \mathbf{Z}_b = \boldsymbol{z}; \boldsymbol{\theta})p_a(\boldsymbol{z})p_b(\boldsymbol{z})$.

E.3 Proofs for VAE with Disentanglement Priors

The main difficulty of maximum likelihood learning for the optimization problem (2) is that the marginal probability of data $p(\mathbf{X}|C, y)$ under the model is intractable. We apply the variational techniques to derive the ELBO for the optimization problem (2), whose constraint is removed by introducing the disentanglement priors.

In the VAE framework, we adopt variational distributions to approximate true distributions (Kingma and Welling, 2019), which ends up maximizing an ELBO. More specifically, we introduce a variational posterior $q_\phi(\mathbf{Z}_c, \mathbf{Z}_y|\mathbf{X}, C, y)$ to approximate the true posterior $p_{\boldsymbol{\theta}}(\mathbf{Z}_c, \mathbf{Z}_y|\mathbf{X}, C, y)$, and derive the ELBO for $p_{\boldsymbol{\theta}}(\mathbf{X}|C, y)$ in Sec. E.3.1:

$$\begin{aligned}&\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{Z}_c,\mathbf{Z}_y|X,C,y)}[\log p_{\boldsymbol{\theta}}(X, \mathbf{Z}_c, \mathbf{Z}_y|C, y)\\ &- \log q_{\boldsymbol{\phi}}(\mathbf{Z}_c, \mathbf{Z}_y|X, C, y)]\end{aligned} \tag{8}$$

We show in Sec. E.3.2 that the ELBO objective is further decomposed into:

$$\begin{aligned}&\overbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{Z}_c,\mathbf{Z}_y|\mathbf{X},C,y)}[\log p_{\boldsymbol{\theta}}(\mathbf{X}|\mathbf{Z}_c, \mathbf{Z}_y)]}^{\mathcal{L}_r}\\ &- \mathbb{D}_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{Z}_c|\mathbf{X}, C)\|p_{\boldsymbol{\theta}}(\mathbf{Z}_c|C))\\ &- \mathbb{D}_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{Z}_y|\mathbf{X}, y)\|p_{\boldsymbol{\theta}}(\mathbf{Z}_y|y))\end{aligned} \tag{9}$$

where the first term is referred to as the reconstruction loss $\mathcal{L}_r$, the other terms constitute regularizers.

E.3.1 Evidence lower bound (ELBO)

$$\log p(\mathbf{X}) \geq \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{Z}_c,\mathbf{Z}_y|\mathbf{X},C,y)}[\log p_{\boldsymbol{\theta}}(\mathbf{X}, \mathbf{Z}_c, \mathbf{Z}_y|C, y) - \log q_{\boldsymbol{\phi}}(\mathbf{Z}_c, \mathbf{Z}_y|\mathbf{X}, C, y)]$$

Proof:

$$\begin{aligned}
&\mathbb{D}_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{Z}_c, \mathbf{Z}_y|\mathbf{X}, C, y)\|p_{\boldsymbol{\theta}}(\mathbf{Z}_c, \mathbf{Z}_y|\mathbf{X}, C, y)) \\
=&-\int q_{\boldsymbol{\phi}}(\mathbf{Z}_c, \mathbf{Z}_y|\mathbf{X}, \boldsymbol{\alpha}_c, \boldsymbol{\beta}_y)\Big[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{Z}_c, \mathbf{Z}_y|\mathbf{X}, C, y))}{q_{\boldsymbol{\phi}}(\mathbf{Z}_c, \mathbf{Z}_y|\mathbf{X}, \boldsymbol{\alpha}_c, \boldsymbol{\beta}_y)}\Big]d\mathbf{Z}_c d\mathbf{Z}_y \\
=&-\int q_{\boldsymbol{\phi}}(\mathbf{Z}_c, \mathbf{Z}_y|\mathbf{X}, \boldsymbol{\alpha}_c, \boldsymbol{\beta}_y)\Big[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{X}, \mathbf{Z}_c, \mathbf{Z}_y|C, y))}{q_{\boldsymbol{\phi}}(\mathbf{Z}_c, \mathbf{Z}_y|\mathbf{X}, \boldsymbol{\alpha}_c, \boldsymbol{\beta}_y)p_{\boldsymbol{\theta}}(\mathbf{X}|\boldsymbol{\alpha}_c, \boldsymbol{\beta}_y)}\Big]d\mathbf{Z}_c d\mathbf{Z}_y \\
=&-\int q_{\boldsymbol{\phi}}(\mathbf{Z}_c, \mathbf{Z}_y|\mathbf{X}, \boldsymbol{\alpha}_c, \boldsymbol{\beta}_y)\Big[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{X}, \mathbf{Z}_c, \mathbf{Z}_y|C, y))}{q_{\boldsymbol{\phi}}(\mathbf{Z}_c, \mathbf{Z}_y|\mathbf{X}, \boldsymbol{\alpha}_c, \boldsymbol{\beta}_y)} - \log p_{\boldsymbol{\theta}}(\mathbf{X}|\boldsymbol{\alpha}_c, \boldsymbol{\beta}_y)\Big]d\mathbf{Z}_c d\mathbf{Z}_y
\end{aligned}$$

$\mathbb{D}_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{Z}_c, \mathbf{Z}_y|\mathbf{X}, C, y)\|p_{\boldsymbol{\theta}}(\mathbf{Z}_c, \mathbf{Z}_y|\mathbf{X}, C, y)) \geq 0$, therefore:

$$\begin{aligned}
\log p_{\boldsymbol{\theta}}(\mathbf{X}|\boldsymbol{\alpha}_c, \boldsymbol{\beta}_y) \geq& \int q_{\boldsymbol{\phi}}(\mathbf{Z}_c, \mathbf{Z}_y|\mathbf{X}, \boldsymbol{\alpha}_c, \boldsymbol{\beta}_y)\Big[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{X}, \mathbf{Z}_c, \mathbf{Z}_y|C, y))}{q_{\boldsymbol{\phi}}(\mathbf{Z}_c, \mathbf{Z}_y|\mathbf{X}, \boldsymbol{\alpha}_c, \boldsymbol{\beta}_y)}\Big]d\mathbf{Z}_c d\mathbf{Z}_y \\
=&\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{Z}_c, \mathbf{Z}_y|\mathbf{X}, C, y)}[\log p_{\boldsymbol{\theta}}(\mathbf{X}, \mathbf{Z}_c, \mathbf{Z}_y|C, y) - \log q_{\boldsymbol{\phi}}(\mathbf{Z}_c, \mathbf{Z}_y|\mathbf{X}, C, y)]
\end{aligned}$$

E.3.2 ELBO decomposition

If $\mathbf{Z}_c \perp\!\!\!\perp \mathbf{Z}_y|C$, then

$$\text{ELBO} = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{Z}_c, \mathbf{Z}_y|\mathbf{X}, C, y)}[\log p_{\boldsymbol{\theta}}(\mathbf{X}|\mathbf{Z}_c, \mathbf{Z}_y)] - \mathbb{D}_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{Z}_c|\mathbf{X}, C)\|p_{\boldsymbol{\theta}}(\mathbf{Z}_c|C)) - \mathbb{D}_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{Z}_y|\mathbf{X}, y)\|p_{\boldsymbol{\theta}}(\mathbf{Z}_y|y)).$$

Proof:

Let $p_{\boldsymbol{\theta}}(\mathbf{X}, \mathbf{Z}_c, \mathbf{Z}_y|C, y) = p_{\boldsymbol{\theta}}(\mathbf{X}|\mathbf{Z}_c, \mathbf{Z}_y)p_{\boldsymbol{\theta}}(\mathbf{Z}_c|C)p_{\boldsymbol{\theta}}(\mathbf{Z}_y|y)$, then

$$\begin{aligned}
&\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{Z}_c, \mathbf{Z}_y|\mathbf{X}, C, y)}[\log p_{\boldsymbol{\theta}}(\mathbf{X}, \mathbf{Z}_c, \mathbf{Z}_y|C, y) - \log q_{\boldsymbol{\phi}}(\mathbf{Z}_c, \mathbf{Z}_y|\mathbf{X}, C, y)] \\
=&\int q_{\boldsymbol{\phi}}(\mathbf{Z}_c, \mathbf{Z}_y|\mathbf{X}, \boldsymbol{\alpha}_c, \boldsymbol{\beta}_y)\Big[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{X}|\mathbf{Z}_c, \mathbf{Z}_y)p_{\boldsymbol{\theta}}(\mathbf{Z}_c|C)p_{\boldsymbol{\theta}}(\mathbf{Z}_y|y))}{q_{\boldsymbol{\phi}}(\mathbf{Z}_c, \mathbf{Z}_y|\mathbf{X}, \boldsymbol{\alpha}_c, \boldsymbol{\beta}_y)}\Big]d\mathbf{Z}_c d\mathbf{Z}_y \\
=&\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{Z}_c, \mathbf{Z}_y|\mathbf{X}, C, y)}[\log p_{\boldsymbol{\theta}}(\mathbf{X}|\mathbf{Z}_c, \mathbf{Z}_y)] + \int q_{\boldsymbol{\phi}}(\mathbf{Z}_c, \mathbf{Z}_y|\mathbf{X}, \boldsymbol{\alpha}_c, \boldsymbol{\beta}_y)\Big[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{Z}_c|C)p_{\boldsymbol{\theta}}(\mathbf{Z}_y|y))}{q_{\boldsymbol{\phi}}(\mathbf{Z}_c, \mathbf{Z}_y|\mathbf{X}, \boldsymbol{\alpha}_c, \boldsymbol{\beta}_y)}\Big]d\mathbf{Z}_c d\mathbf{Z}_y \\
=&\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{Z}_c, \mathbf{Z}_y|\mathbf{X}, C, y)}[\log p_{\boldsymbol{\theta}}(\mathbf{X}|\mathbf{Z}_c, \mathbf{Z}_y)] + \int q_{\boldsymbol{\phi}}(\mathbf{Z}_c, \mathbf{Z}_y|\mathbf{X}, C, y)\Big[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{Z}_c|C)p_{\boldsymbol{\theta}}(\mathbf{Z}_y|y))}{q_{\boldsymbol{\phi}}(\mathbf{Z}_c|\mathbf{Z}_y, \mathbf{X}, C)q_{\boldsymbol{\phi}}(\mathbf{Z}_y|\mathbf{X}, y)}\Big]d\mathbf{Z}_c d\mathbf{Z}_y
\end{aligned}$$

Because $\mathbf{Z}_c \perp\!\!\!\perp \mathbf{Z}_y|C$, $q_{\boldsymbol{\phi}}(\mathbf{Z}_c, \mathbf{Z}_y|\mathbf{X}, C, y) = q_{\boldsymbol{\phi}}(\mathbf{Z}_c|\mathbf{X}, C)q_{\boldsymbol{\phi}}(\mathbf{Z}_y|\mathbf{X}, y)$, thus

$$\begin{aligned}
&\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{Z}_c, \mathbf{Z}_y|\mathbf{X}, C, y)}[\log p_{\boldsymbol{\theta}}(\mathbf{X}|\mathbf{Z}_c, \mathbf{Z}_y)] + \int q_{\boldsymbol{\phi}}(\mathbf{Z}_c|\mathbf{X}, C)q_{\boldsymbol{\phi}}(\mathbf{Z}_y|\mathbf{X}, y)\Big[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{Z}_c|C)p_{\boldsymbol{\theta}}(\mathbf{Z}_y|y))}{q_{\boldsymbol{\phi}}(\mathbf{Z}_c|\mathbf{X}, C)q_{\boldsymbol{\phi}}(\mathbf{Z}_y|\mathbf{X}, y)}\Big]d\mathbf{Z}_c d\mathbf{Z}_y \\
=&\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{Z}_c, \mathbf{Z}_y|\mathbf{X}, C, y)}[\log p_{\boldsymbol{\theta}}(\mathbf{X}|\mathbf{Z}_c, \mathbf{Z}_y)] - \mathbb{D}_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{Z}_c|\mathbf{X}, C)q_{\boldsymbol{\phi}}(\mathbf{Z}_y|\mathbf{X}, y)\|p_{\boldsymbol{\theta}}(\mathbf{Z}_c|C)p_{\boldsymbol{\theta}}(\mathbf{Z}_y|y)) \\
=&\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{Z}_c, \mathbf{Z}_y|\mathbf{X}, C, y)}[\log p_{\boldsymbol{\theta}}(\mathbf{X}|\mathbf{Z}_c, \mathbf{Z}_y)] - \mathbb{D}_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{Z}_c|\mathbf{X}, C)\|p_{\boldsymbol{\theta}}(\mathbf{Z}_c|C)) - \mathbb{D}_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{Z}_y|\mathbf{X}, y)\|p_{\boldsymbol{\theta}}(\mathbf{Z}_y|y)).
\end{aligned}$$

Note that, the last step is derived by applying the chain rule of KL divergence.

E.3.3 Derivation of the regularization term for latent label representations

If $q_{\boldsymbol{\phi}}(\mathbf{Z}_y|\mathbf{X}, y) = \mathcal{N}(\mathbf{Z}_y; \boldsymbol{\mu}_y^q, \mathrm{diag}(\boldsymbol{\sigma}^2)_y)$ and $p_{\boldsymbol{\theta}}(\mathbf{Z}_y|y) = \mathcal{N}(\mathbf{Z}_y; \boldsymbol{\mu}_y^p, \lambda_y\mathbf{I})$, where $\boldsymbol{\mu}_y^q, \log\boldsymbol{\sigma}_y = \text{LabelEncoder}(\mathbf{X})$ and $\boldsymbol{\mu}_y^p = \mathbf{W}_y\Phi(l)$, then we have:

$$\mathbb{D}_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{Z}_y|\mathbf{X}, y)\|p_{\boldsymbol{\theta}}(\mathbf{Z}_y|y)) = \frac{1}{2\lambda_y}\|\mathbf{Z}_y - \boldsymbol{\mu}_y^p\|^2 - \log\boldsymbol{\sigma}_y^q + \text{const}$$

Proof: Let $\mathbf{Z}_y = \boldsymbol{\mu}_y^q + \boldsymbol{\sigma}_y \odot \boldsymbol{\epsilon}_y$, where $\boldsymbol{\epsilon}_y$ is drawn from $\mathcal{N}(0, \mathbf{I})$.

$$\begin{aligned}
\mathbb{D}_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{Z}_y|\mathbf{X}, y)\|p_{\boldsymbol{\theta}}(\mathbf{Z}_y|y)) =& \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{Z}_y|\mathbf{X}, y)}\big[\log q_{\boldsymbol{\phi}}(\mathbf{Z}_y|\mathbf{X}, y) - \log p_{\boldsymbol{\theta}}(\mathbf{Z}_y|y)\big] \\
=& \mathbb{E}_{p(\boldsymbol{\epsilon}_y)}\big[\log q_{\boldsymbol{\phi}}(\mathbf{Z}_y|\mathbf{X}, y) - \log p_{\boldsymbol{\theta}}(\mathbf{Z}_y|y)\big]
\end{aligned}$$

$p_{\boldsymbol{\theta}}(\mathbf{Z}_y|y) = \mathcal{N}(\mathbf{Z}_y; \boldsymbol{\mu}_y^p, \lambda_y\mathbf{I})$, thus

$$\log p_{\boldsymbol{\theta}}(\mathbf{Z}_y|y) = -\frac{1}{2\lambda_y}\|\mathbf{Z}_y - \boldsymbol{\mu}_y^p\|^2 + \text{const}$$

Using the reparameterization trick,

$$\begin{aligned}
\log q_{\boldsymbol{\phi}}(\mathbf{Z}_y|\mathbf{X}, y) =& \log p(\boldsymbol{\epsilon}_y) - \log|\det(\frac{\partial\mathbf{Z}_y}{\partial\boldsymbol{\epsilon}_y})| \\
=& \log\mathcal{N}(\boldsymbol{\epsilon}_y; 0, \mathbf{I}) - \log\boldsymbol{\sigma}_y^q
\end{aligned}$$

Put them together

$$\mathbb{E}_{p(\boldsymbol{\epsilon}_y)}\big[\log q_{\boldsymbol{\phi}}(\mathbf{Z}_y|\mathbf{X}, y) - \log p_{\boldsymbol{\theta}}(\mathbf{Z}_y|y)\big] = \frac{1}{2\lambda_y}\|\mathbf{Z}_y - \boldsymbol{\mu}_y^p\|^2 - \log\boldsymbol{\sigma}_y^q + \text{const}$$

E.3.4 Derivation of the regularization term for latent content representations

We assume $q_{\boldsymbol{\phi}}(\mathbf{Z}_c|\mathbf{X}, C) = \mathcal{N}(\mathbf{Z}_c; \boldsymbol{\mu}_c^q, \text{diag}(\boldsymbol{\sigma}_c^2))$ and $p_{\boldsymbol{\theta}}(\mathbf{Z}_c|C) = \sum_{k=1}^K p(M=k)\mathcal{N}(\mathbf{Z}_c; \boldsymbol{\mu}_{c,k}^p, \lambda_c \mathbf{I})$ then we have

$$\mathbb{D}_{\text{KL}}(q_{\boldsymbol{\phi}}(\mathbf{Z}_c|\mathbf{X}, C)\|p_{\boldsymbol{\theta}}(\mathbf{Z}_c|C)) = \sum_{k=1}^K p(M=k|\mathbf{Z}_c)\big[\frac{1}{2\lambda_c}\|\mathbf{Z}_c - \boldsymbol{\mu}_{c,k}^p\|^2\big] - \log \boldsymbol{\sigma}_c + \text{const}$$

Proof:

Let $\mathbf{Z}_c = \boldsymbol{\mu}_c^q + \boldsymbol{\sigma}_c \odot \boldsymbol{\epsilon}_c$, where $\boldsymbol{\epsilon}_c$ is drawn from $\mathcal{N}(0, \mathbf{I})$.

$$\begin{aligned}\mathbb{D}_{\text{KL}}(q_{\boldsymbol{\phi}}(\mathbf{Z}_c|\mathbf{X}, C)\|p_{\boldsymbol{\theta}}(\mathbf{Z}_c|C)) =& \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{Z}_c|\mathbf{X},C)}\big[\log q_{\boldsymbol{\phi}}(\mathbf{Z}_c|\mathbf{X}, C) - \log p_{\boldsymbol{\theta}}(\mathbf{Z}_c|C)\big] \\ =& \mathbb{E}_{p(\boldsymbol{\epsilon}_c)}\big[\log q_{\boldsymbol{\phi}}(\mathbf{Z}_c|\mathbf{X}, C) - \log p_{\boldsymbol{\theta}}(\mathbf{Z}_c|C)\big] \\ =& \mathbb{E}_{p(\boldsymbol{\epsilon}_c)}\big[\log q_{\boldsymbol{\phi}}(\mathbf{Z}_c|\mathbf{X}, C)\big] - \log p_{\boldsymbol{\theta}}(\mathbf{Z}_c|C)\end{aligned}$$

Using the reparameterization trick,

$$\mathbb{E}_{p(\boldsymbol{\epsilon}_c)}\big[\log q_{\boldsymbol{\phi}}(\mathbf{Z}_c|\mathbf{X}, y)\big] = -\log \boldsymbol{\sigma}_c^q$$

It remains to estimate $\log p_{\boldsymbol{\theta}}(\mathbf{Z}_c|C)$, which is a Gaussian mixture. Let $\gamma_k \in \{0, 1\}$ indicate the kth component of $\boldsymbol{z}$, the likelihood function for $\boldsymbol{z}$ takes the form

$$p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{\gamma}) = \prod_{k=1}^K p(M=k)^{\gamma_k}\mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}_{c,k}^p, \lambda_c\mathbf{I})^{\gamma_k}$$

In the work, we consider using EM (Bishop and Nasrabadi, 2006), which estimates the expected value of the complete log likelihood function given by

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\gamma}}\big[\log p_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{\gamma})\big] =& \sum_{k=1}^K \mathbb{E}(\gamma_k)\{\log p(M=k) + \log \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}_{c,k}^p, \lambda_c\mathbf{I})\} \\ =& \sum_{k=1}^K \mathbb{E}(\gamma_k)\{-\frac{1}{2\lambda_c}\|\mathbf{Z}_c - \boldsymbol{\mu}_{c,k}^p\|^2\}\} + \text{const} \qquad (10)\end{aligned}$$

where $\mathbb{E}(\gamma_k) = p(M=k|\boldsymbol{z}_c) = \frac{p(M=k)\mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}_{c,k}^p, \lambda_c\mathbf{I})}{\sum_{j=1}^K p(M=j)\mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}_{c,j}^p, \lambda_c\mathbf{I})}$, estimated in the E-step.

For hard EM:

E-step. For each latent content representation $\boldsymbol{z}_c$, the most likely component Gaussian is given by $k^* = \arg\max_k p_{\boldsymbol{\theta}}(M=k)\mathcal{N}(\mathbf{Z}_c; \boldsymbol{\mu}_{c,k}^p, \lambda_c\mathbf{I})$.

M-step. Put the estimated k^* into Eq. (10), this step aims to optimize

$$\sum_{k=1}^K \gamma_{k^*}\{-\frac{1}{2\lambda_c}\|\mathbf{Z}_c - \boldsymbol{\mu}_{c,k}^p\|^2\} + \text{const}$$

Put them together, we have

$$\mathbb{E}_{p(\boldsymbol{\epsilon}_c)}\big[\log q_{\boldsymbol{\phi}}(\mathbf{Z}_c|\mathbf{X}, C)\big] - \log p_{\boldsymbol{\theta}}(\mathbf{Z}_c|C) = \sum_{k=1}^K p(M=k|\mathbf{Z}_c)\big[\frac{1}{2\lambda_c}\|\mathbf{Z}_c - \boldsymbol{\mu}_{c,k}^p\|^2\big] - \log \boldsymbol{\sigma}_c + \text{const}$$

F Model Regularization

F.1 HSIC Regularization

In each batch, the model collects the latent representations $(\mathbf{Z}_c, \mathbf{Z}_y)$, which are a content representation matrix and a label representation matrix respectively. We apply the linear kernel to build a Gram matrix $\mathbf{K}_c = \mathbf{Z}_c\mathbf{Z}_c^\intercal$ for content and a Gram matrix $\mathbf{K}_y = \mathbf{Z}_y\mathbf{Z}_y^\intercal$ for labels. The HSIC metric is computed as

$$\text{HSIC}(\mathbf{Z}_c, \mathbf{Z}_y) = \frac{1}{m^2}\text{trace}(\mathbf{K}_c\mathbf{H}\mathbf{K}_y\mathbf{H}) \qquad (11)$$

where $\mathbf{H} = \mathbf{I} - \frac{1}{m}\mathbf{1}\mathbf{1}^\intercal$ and m is the size of the batch. Alternatively, we can try the Gaussian Kernel for both types of representations.

F.2 MMD Regularization

The MMD divergence is given by (Gretton et al., 2012):

$$\text{MMD}(Z_c, Z_y) \;=\; \frac{1}{m^2}\sum_{i=0}^{m}\sum_{j=0}^{m} k(\boldsymbol{z}_i^c, \boldsymbol{z}_j^c) \;-\; \frac{2}{m^2}\sum_{i=0}^{m}\sum_{j=0}^{m} k(\boldsymbol{z}_i^c, \boldsymbol{z}_j^y) \;+\; \frac{1}{m^2}\sum_{i=0}^{m}\sum_{j=0}^{m} k(\boldsymbol{z}_i^y, \boldsymbol{z}_j^y) \quad (12)$$

where $k(\cdot, \cdot)$ is a kernel function, whereby we choose the linear kernel in our experiments. Maximizing MMD increases the similarity of the latent representations of the same type, while decreases the similarity of the latent representations across types.