ControlVAE: Controllable Variational Autoencoder

Huajie Shao [1] **Shuochao Yao** [1] **Dachun Sun** [1] **Aston Zhang** [2]
Shengzhong Liu [1] **Dongxin Liu** [1] **Jun Wang** [3] **Tarek Abdelzaher** [1]

Abstract

Variational Autoencoders (VAE) and their variants have been widely used in a variety of applications, such as dialog generation, image generation and disentangled representation learning. However, the existing VAE models have some limitations in different applications. For example, a VAE easily suffers from KL vanishing in language modeling and low reconstruction quality for disentangling. To address these issues, we propose a novel controllable variational autoencoder framework, ControlVAE, that combines a controller, inspired by automatic control theory, with the basic VAE to improve the performance of resulting generative models. Specifically, we design a new non-linear PI controller, a variant of the proportional-integral-derivative (PID) control, to automatically tune the hyperparameter (weight) added in the VAE objective using the output KL-divergence as feedback during model training. The framework is evaluated using three applications; namely, language modeling, disentangled representation learning, and image generation. The results show that ControlVAE can achieve much better reconstruction quality than the competitive methods for the comparable disentanglement performance. For language modelling, it not only averts the KL-vanishing, but also improves the diversity of generated text. Finally, we also demonstrate that ControlVAE improves the reconstruction quality for image generation compared to the original VAE.

1. Introduction

This paper proposes a novel controllable variational autoencoder, ControlVAE [1], that leverages automatic control to precisely control the trade-off between data reconstruction accuracy bounds (from a learned latent representation) and application-specific constraints, such as output diversity or disentangled latent factor representation. Specifically, a controller is designed that stabilizes the value of KL-divergence (between the learned approximate distribution of the latent variables and their true distribution) in the VAE's objective function to achieve the desired trade-off, thereby improving application-specific performance metrics of several existing VAE models.

The work is motivated by the increasing popularity of VAEs as an unsupervised generative modeling framework that learns an approximate mapping between Gaussian latent variables and data samples when the true latent variables have an intractable posterior distribution (Sohn et al., 2015; Kingma & Welling, 2013). Since VAEs can directly work with both continuous and discrete input data (Kingma & Welling, 2013), they have been widely adopted in various applications, such as image generation (Yan et al., 2016; Liu et al., 2017), dialog generation (Wang et al., 2019; Hu et al., 2017), and disentangled representation learning (Higgins et al., 2017; Kim & Mnih, 2018).

Popular VAE applications often involve a trade-off between reconstruction accuracy bounds and some other application-specific goal, effectively manipulated through KL-divergence. For example, in (synthetic) text or image generation, a goal is to produce *new original text or images*, as opposed to reproducing one of the samples in training data. In text generation, if KL-divergence is too low, output diversity is reduced (Bowman et al., 2015), which is known as the KL-vanishing problem. To increase output diversity, it becomes advantageous to artificially *increase KL-divergence*. The resulting approximation was shown to produce more diverse, yet still authentic-looking outputs. Conversely, disentangled representation learning (Denton et al., 2017) leverages the observation that KL-divergence in the VAE constitutes an upper bound on information transfer

[1]Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, USA. [2]AWS Deep Learning, CA, USA. [3]Alibaba Group at Seattle, Washington, USA. Correspondence to: Tarek Abdelzaher <zaher@illinois.edu>, Huajie Shao <hshao5@illinois.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 108, 2020.

[1]Source code is publicly available at https://github.com/shj1987/ControlVAE-ICML2020.git

through the latent channels per data sample (Burgess et al., 2018). Artificially *decreasing KL-divergence* (e.g., by increasing its weight in a VAE's objective function, which is known as the β-VAE) therefore imposes a stricter information bottleneck, which was shown to force the learned latent factors to become more independent (i.e., non-redundant), leading to a better disentangling. The above examples suggest that a useful extension of VAEs is one that allows users to exercise explicit control over KL-divergence in the objective function. ControlVAE realizes this extension.

We apply ControlVAE to three different applications: language modeling, disentangling, and image generation. Evaluation results on real-world datasets demonstrate that ControlVAE is able to achieve an *adjustable trade-off* between reconstruction error and KL-divergence. It can discover more disentangled factors and significantly reduce the reconstruction error compared to the β-VAE (Burgess et al., 2018) for disentangling. For language modeling, it can not only completely avert the KL vanishing problem, but also improve the diversity of generated data. Finally, we also show that ControlVAE improves the reconstruction quality on image generation task via slightly increasing the value of KL divergence compared with the original VAE.

2. Preliminaries

The objective function of VAEs consists of two terms: log-likelihood and KL-divergence. The first term tries to reconstruct the input data, while KL-divergence has the desirable effect of keeping the representation of input data sufficiently diverse. In particular, KL-divergence can affect both the reconstruction quality and diversity of generated data. If the KL-divergence is too high, it would affect the accuracy of generated samples. If it is too low, output diversity is reduced, which may be a problem in some applications such as language modeling (Bowman et al., 2015) (where it is known as the KL-vanishing problem).

To mitigate KL vanishing, one promising way is to add an extra hyperparameter $\beta (0 \leq \beta \leq 1)$ in the VAE objective function to control the KL-divergence via increasing β from 0 until to 1 with sigmoid function or cyclic function (Liu et al., 2019). These methods, however, blindly change β without sampling the actual KL-divergence during model training. Using a similar methodology, researchers recently developed a new β-VAE ($\beta > 1$) (Higgins et al., 2017; Burgess et al., 2018) to learn the disentangled representations by controlling the value of KL-divergence. However, β-VAE suffers from high reconstruction errors (Kim & Mnih, 2018), because it adds a very large β in the VAE objective so the model tends to focus disproportionately on optimizing the KL term. In addition, its hyperparameter is fixed during model training, missing the chance of balancing the reconstruction error and KL-divergence.

The core technical challenge responsible for the above application problems lies in the difficulty to tune the weight of the KL-divergence term during model training. Inspired by control systems, we fix this problem using feedback control. Our controllable variational autoencoder is illustrated in Fig. 1. It samples the output KL-divergence at each training step t, and feeds it into an algorithm that tunes the hyperparameter, $\beta(t)$, accordingly, aiming to stabilize KL-divergence at a desired value, called the *set point*.

We further design a non-linear PI controller, a variant of the PID control algorithm (Åström et al., 2006), to tune the hyperparameter $\beta(t)$. PID control is the basic and most prevalent form of feedback control in a large variety of industrial (Åström et al., 2006) and software performance control (Hellerstein et al., 2004) applications. The general model of PID controller is defined by

$$\beta(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}, \quad (1)$$

where $\beta(t)$ is the output of the controller; $e(t)$ is the error between the actual value and the desired value at time t; K_p, K_i and K_d denote the coefficients for the P term, I term and D term, respectively.

The basic idea of the PID algorithm is to calculate an error, $e(t)$, between a set point (in this case, the desired KL-divergence) and the current value of the controlled variable (in this case, the actual KL-divergence), then apply a correction in a direction that reduces that error. The correction is applied to some intermediate directly accessible variable (in our case, $\beta(t)$) that influences the value of the variable we ultimately want to control (KL-divergence). In general, the correction computed by the controller is the weighted sum of three terms; one changes with error (P), one changes with the integral of error (I), and one changes with the derivative of error (D). In a nonlinear controller, the changes can be described by *nonlinear* functions. Note that, since derivatives essentially compute the slope of a signal, when the signal is noisy, the slope often responds more to variations induced by noise. Hence, following established best practices in control of noisy systems, we do not use the derivative (D) term in our specific controller. Next, we introduce VAEs and our objective in more detail.

2.1. The Variational Autoencoder (VAE)

Suppose that we have a dataset $\mathbf{x}$ of n i.i.d. samples that are generated by the ground-truth latent variable $\mathbf{z}$, interpreted as the representation of the data. Let $p_\theta(\mathbf{x}|\mathbf{z})$ denote a probabilistic *decoder* with a neural network to generate data $\mathbf{x}$ given the latent variable $\mathbf{z}$. The distribution of representation corresponding to the dataset $\mathbf{x}$ is approximated by the variational posterior, $q_\phi(\mathbf{z}|\mathbf{x})$, which is produced by an *encoder* with a neural network. The Variational Autoencoder

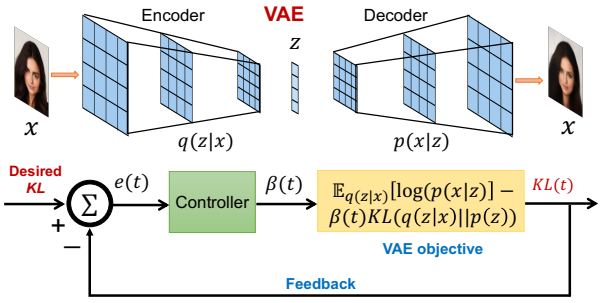


Figure 1. Framework of ControlVAE. It combines a controller with the basic VAE framework to stabilize the KL divergence to a specified value via automatically tuning the weight $\beta(t)$ in the objective.

(VAE) (Rezende et al., 2014; Kingma & Welling, 2013) has been one of the most popular generative models. The basic idea of VAE can be summarized in the following: (1) VAE encodes the input data samples $\mathbf{x}$ into a latent variable $\mathbf{z}$ as its distribution of representation via a probabilistic encoder, which is parameterised by a neural network. (2) then adopts the decoder to reconstruct the original input data based on the samples from $\mathbf{z}$. VAE tries to maximize the marginal likelihood of the reconstructed data, but it involves intractable posterior inference. Thus, researchers adopt backpropagation and stochastic gradient descent (Kingma & Welling, 2013) to optimize its variational lower bound of log likelihood (Kingma & Welling, 2013).

$$\begin{aligned}&\log p_\theta(\mathbf{x}) \geq \mathcal{L}_{vae} \\ &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})),\end{aligned} \tag{2}$$

where $p(\mathbf{z})$ is the prior distribution, e.g., Unit Gaussian. $p_\theta(\mathbf{x}|\mathbf{z})$ is a probabilistic *decoder* parameterized by a neural network to generate data $\mathbf{x}$ given the latent variable $\mathbf{z}$, and the posterior distribution of latent variable $\mathbf{z}$ given data $\mathbf{x}$ is approximated by the variational posterior, $q_\phi(\mathbf{z}|\mathbf{x})$, which is parameterized by an *encoder* network. The VAE is trained by maximizing $\mathcal{L}_{vae}$, which consists of a reconstruction term and a KL term, over the training data.

However, the basic VAE models cannot explicitly control the KL-divergence to a specified value. They also often suffer from KL vanishing (in language modeling (Bowman et al., 2015; Liu et al., 2019)), which means the KL-divergence becomes zero during optimization.

2.2. β-VAE

β-VAE (Higgins et al., 2017; Chen et al., 2018a) is an extension to the basic VAE framework, often used as an unsupervised method for learning a disentangled representation of the data generative factors. A disentangled representation, according to the literature (Bengio et al., 2013), is defined as one where single latent units are sensitive to changes in single generative factors, while being relatively invariant to changes in other factors. Compared to the original VAE, β-VAE adds an extra hyperparameter $\beta(\beta > 1)$ as a weight of KL-divergence in the original VAE objective (2). It can be expressed by

$$\mathcal{L}_\beta = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \beta D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})). \tag{3}$$

In order to discover more disentangled factors, researchers further put a constraint on total information capacity, C, to control the capacity of the information bottleneck (KL-divergence) (Burgess et al., 2018). Then Lagrangian method is adopted to solve the following optimization problem.

$$\mathcal{L}_\beta = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \beta|D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) - C|, \tag{4}$$

where β is a large hyperparameter (e.g., 100).

However, one drawback of β-VAE is that it obtains good disentangling at the cost of reconstruction quality. When the weight β is large, the optimization algorithm tends to optimize the second term in (4), leading to a high reconstruction error.

The above background suggests that a common challenge in applying VAEs (and their extensions) lies in appropriate weight allocation among the reconstruction accuracy and KL-divergence in the VAEs objective function. As mentioned earlier, we solve this using a nonlinear PI controller that manipulates the value of the non-negative hyperparameter, $\beta(t)$. This algorithm is described next.

3. The ControlVAE Algorithm

During model training, we sample the output KL-divergence, which we denote by $\hat{v}_{kl}(t)$, at training step t. The sampled KL-divergence is then compared to the set point, v_{kl}, and the difference, $e(t) = v_{kl} - \hat{v}_{kl}(t)$ then used as the feedback to a controller to calculate the hyperparameter $\beta(t)$. ControlVAE can be expressed by the following variational lower bound:

$$\mathcal{L} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \beta(t) D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})), \tag{5}$$

When KL-divergence drops below the set point, the controller counteracts this change by reducing the hyperparameter $\beta(t)$ (to reduce penalty for KL-divergence in the objective function (5)). The reduced weight, $\beta(t)$, allows KL-divergence to grow, thus approaching the set point again. Conversely, when KL-divergence grows above the set point, the controller increases $\beta(t)$ (up to a certain value), thereby increasing the penalty for KL-divergence and forcing it to decrease. This effect is achieved by computing $\beta(t)$ using Equation (6), below, which is an instance of nonlinear PI control:

$$\beta(t) = \frac{K_p}{1 + \exp(e(t))} - K_i \sum_{j=0}^{t} e(j) + \beta_{min}, \tag{6}$$

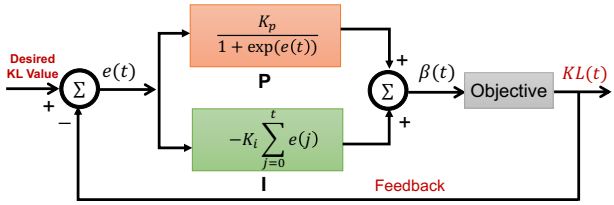


Figure 2. PI controller. It uses the output KL-divergence at training step t as the feedback to the PI algorithm to compute $\beta(t)$.

where K_p and K_i are the constants. The first term (on the right hand side) ranges between 0 and K_p thanks to the exponential function $exp(.)$. Note that when error is large and positive (KL-diverge is below set point), the first term approaches 0, leading to a lower $\beta(t)$ that encourages KL-divergence to grow. Conversely, when error is large and negative (KL-divergence above set point), the first term approaches its maximum (which is K_p), leading to a higher $\beta(t)$ that encourages KL-divergence to shrink.

The second term of the controller sums (integrates) past errors with a sampling period T (one training step in this paper). This creates a progressively stronger correction (until the sign of the error changes). The negative sign ensures that while errors remain positive (i.e., when KL-divergence is below set point), this term continues to decrease, whereas while errors remain negative (i.e., when KL-divergence is above set point), this term continues to increase. In both cases, the change forces $\beta(t)$ in a direction that helps KL-divergence approach the set point. In particular, note that when the error becomes zero, the second term (and thus the entire right hand side) stops changing, allowing controller output, $\beta(t)$, to stay at the same value that hopefully caused the zero error in the first place. This allows the controller to "lock in" the value of $\beta(t)$ that meets the KL-divergence set point. Finally, β_{min} is an application-specific constant. It effectively shifts the range within which $\beta(t)$ is allowed to vary. This PI controller is illustrated in Fig. 2.

3.1. PI Parameter Tuning for ControlVAE

One challenge of applying the PI control algorithm lies how to tune its parameters, K_p and K_i effectively. While optimal tuning of nonlinear controllers is non-trivial, in this paper we follow a very simple rule: tune these constants to ensure that reactions to errors are sufficiently smooth to allow gradual convergence. Let us first consider the coefficient K_p. Observe that the maximum (positive) error occurs when actual KL-divergence is close to zero. In this case, if v_{kl} is the set point on KL-divergence, then the error, $e(t)$, is approximated by $e(t) \approx v_{kl} - 0 = v_{kl}$. When KL-divergence is too small, the VAE does not learn useful information from input data (Liu et al., 2019). We need to assign $\beta(t)$ a very small non-negative value, so that KL-divergence is encouraged to grow (when the resulting objective function is optimized). In other words, temporarily ignoring other terms in Equation (6), the contribution of the first term alone should be sufficiently small:

$$\frac{K_p}{1+\exp(v_{kl})} \leq \epsilon, \tag{7}$$

where ϵ is a small constant (e.g., 10^{-3} in our implementation). The above (7) can also be rewritten as $K_p \leq (1+\exp(v_{kl}))\epsilon$. Empirically, we find that $K_p = 0.01$ leads to good performance and satisfies the above constraint.

Conversely, when the actual KL-divergence is much larger than the desired value v_{kl}, the error $e(t)$ becomes a large negative value. As a result, the first term in (6) becomes close to a constant, K_p. If the resulting larger value of $\beta(t)$ is not enough to cause KL-divergence to shrink, one needs to gradually continue to increase $\beta(t)$. This is the job of second term. The negative sign in front of that term ensures that when negative errors continue to accumulate, the positive output $\beta(t)$ continues to increase. Since it takes lots of steps to train deep VAE models, the increase per step should be very small, favoring smaller values of K_i. Empirically we found that a value K_i between 10^{-3} and 10^{-4} stabilizes the training. Note that, K_i should not be too small either, because it would then unnecessarily slow down the convergence.

3.2. Set Point Guidelines for ControlVAE

The choice of desired value of KL-divergence (set point) is largely application specific. In general, when $\beta_{min} \leq \beta(t) \leq \beta_{max}$, the upper bound of expected KL-divergence is the value of KL-divergence as ControlVAE converges when $\beta(t) = \beta_{min}$, denoted by V_{max}. Similarly, its lower bound, V_{min}, can be defined as the KL-divergence produced by ControlVAE when $\beta(t) = \beta_{max}$. For feedback control to be most effective (i.e., not run against the above limits), the KL-divergence set point should vary in the range of $[V_{min}, V_{max}]$. Since ControlVAE is an end-to-end learning model, users can customize the desired value of KL-divergence to meet their demand with respect to different applications. For instance, if some users prefer to improve the diversity of text generation and image generation, they can slightly increase the KL-divergence. Otherwise they can reduce the KL-divergence if they want to improve the generation accuracy.

3.3. Summary of the PI Control Algorithm

We summarize the proposed PI control algorithm in Algorithm 1. Our PI algorithm updates the hyperparameter, $\beta(t)$, with the feedback from sampled KL-divergence at training step t. Line 6 computes the error between the desired KL-divergence, $v_{kl}(t)$, and the sampled $\hat{v}_{kl}(t)$. Line 7 to 9

Algorithm 1 PI algorithm.

1: **Input:** desired KL v_{kl}, coefficients K_p, K_i, max/min value β_{max}, β_{min}, iterations N
2: **Output:** hyperparameter $\beta(t)$ at training step t
3: **Initialization**: $I(0) = 0$, $\beta(0) = 0$
4: **for** $t = 1$ **to** N **do**
5: Sample KL-divergence, $\hat{v}_{kl}(t)$
6: $e(t) \leftarrow v_{kl} - \hat{v}_{kl}(t)$
7: $P(t) \leftarrow \frac{K_p}{1+\exp(e(t))}$
8: **if** $\beta_{min} \leq \beta(t-1) \leq \beta_{max}$ **then**
9: $I(t) \leftarrow I(t-1) - K_i e(t)$
10: **else**
11: $I(t) = I(t-1)$ // Anti-windup
12: **end if**
13: $\beta(t) = P(t) + I(t) + \beta_{min}$
14: **if** $\beta(t) > \beta_{max}$ **then**
15: $\beta(t) = \beta_{max}$
16: **end if**
17: **if** $\beta(t) < \beta_{min}$ **then**
18: $\beta(t) = \beta_{min}$
19: **end if**
20: **Return** $\beta(t)$
21: **end for**

calculate the P term and I term for the PI algorithm, respectively. Note that, Line 10 and 11 is a popular constraint in PID/PI design, called anti-windup (Azar & Serrano, 2015; Peng et al., 1996). It effectively disables the integral term of the controller when controller output gets out of range, not to exacerbate the out-of-range deviation. Line 13 is the calculated hyperparameter $\beta(t)$ by PI algorithm in (6). Finally, Line 14 to 19 aim to limit $\beta(t)$ to a certain range, $[\beta_{min}, \beta_{max}]$.

3.4. Applications of ControlVAE

As a preliminary demonstration of the general applicability of the above approach and as an illustration of its customizability, we apply ControlVAE to three different applications stated below.

- **Language modeling**: We first apply ControlVAE to solve the KL vanishing problem meanwhile improve the diversity of generated data. As mentioned in Section 2.1, the VAE models often suffer from KL vanishing in language modeling. The existing methods cannot completely solve the KL vanishing problem or explicitly manipulate the value of KL-divergence. In this paper, we adopt ControlVAE to control KL-divergence to a specified value to avoid KL vanishing using the output KL-divergence. Following PI tuning strategy in Section 3.1, we set K_p, K_i of the PI algorithm in (6) to 0.01 and 0.0001, respectively. In addition, β_{min} is set to 0 and the maximum value of $\beta(t)$ is limited to 1.

- **Disentangling**: We then apply the ControlVAE model to achieve a better trade-off between reconstruction quality and disentangling. As mentioned in Section 2.2, β-VAE ($\beta > 1$) assigns a large hyperparameter to the objective function to control the KL-divergence (information bottleneck), which, however, leads to a large reconstruction error. To mitigate this issue, we adopt ControlVAE to automatically adjust the hyperparameter $\beta(t)$ based on the output KL-divergence during model training. Using the similar methodology in (Burgess et al., 2018), we train a single model by gradually increasing KL-divergence from 0.5 to a desired value C with a step function α for every K training steps. Since $\beta(t) > 1$, we set β_{min} to 1 for the PI algorithm in (6). Following the PI tuning method above, the coefficients K_p and K_i are set to 0.01 and 0.001, respectively.

- **Image generation**: The basic VAE models tend to produce blurry and unrealistic samples for image generation (Zhao et al., 2017a). In this paper, we try to leverage ControlVAE to manipulate (slightly increase) the value of KL-divergence to improve the reconstruction quality for image generation. Different from the original VAE ($\beta(t) = 1$), we extend the range of the hyperparameter, $\beta(t)$, from 0 to 1 in our controlVAE model. Given a desired KL-divergence, controlVAE can automatically tune $\beta(t)$ within that range. For this task, we use the same PI control algorithm and hyperparameters as the above language modeling.

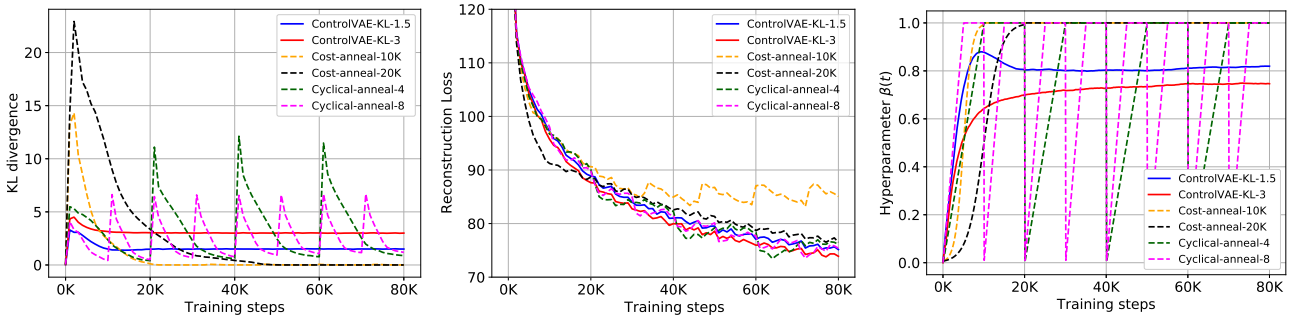
4. Experiments

We evaluate the performance of ControlVAE on benchmark datasets in the three different applications mentioned above.

4.1. Datasets

The datasets used for our experiments are introduced below.

- Language modeling: 1) **Penn Tree Bank (PTB)** (Marcus et al., 1993): it consists of $42,068$ training sentences, $3,370$ validation sentences and $3,761$ testing sentences. 2) **Switchboard(SW)** (Godfrey & Holliman, 1997): it has 2400 two-sided telephone conversations with manually transcribed speech and alignment. The data is randomly split into 2316, 60 and 62 dialog for training, validation and testing.
- Disentangling: 1) **2D Shapes** (Matthey et al., 2017): it has $737,280$ binary 64×64 images of 2D shapes with five ground truth factors (number of values): shape(3), scale(6), orientation(40), x-position(32), y-position(32) (Kim & Mnih, 2018).
- Image generation: 1) **CelebA**(cropped version) (Liu et al., 2015): It has $202,599$ RGB $128 \times 128 \times 3$ images of celebrity faces. The data is split into $192,599$ and $10,000$ images for training and testing.



(a) KL divergence (b) Reconstruction loss (c) $\beta(t)$

Figure 3. Performance comparison for different methods on the PTB data. (a) shows that ControlVAE and Cyclical annealing $(4, 8$ cycles) can avert KL vanishing, while Cost annealing still suffers from KL vanishing after $20K$ and $50K$ training steps. Moreover, ControlVAE can control the KL-divergence and also has lower reconstruction errors than the other methods in (b).

Table 1. Performance comparison for different methods on dialog-generation using SW data over 5 random seeds. Dis-n: higher is better. PPL: lower is better, and self-BLEU lower is better.

Methods/metric	Dis-1	Dis-2	self-BLEU-2	self-BLEU-3	PPL
ControlVAE-KL-35	**6.27K** $\pm$ 41	**95.86K** $\pm$ 1.02K	**0.663** $\pm$ 0.012	**0.447** $\pm$ 0.013	**8.81** $\pm$ 0.05
ControlVAE-KL-25	6.10K $\pm$ 60	83.15K $\pm$ 4.00K	0.698 $\pm$ 0.006	0.495 $\pm$ 0.014	12.47 $\pm$ 0.07
Cost anneal-KL-17	5.71K $\pm$ 87	69.60K $\pm$ 1.53K	0.721 $\pm$ 0.010	0.536 $\pm$ 0.008	16.82 $\pm$ 0.11
Cyclical (KL = 21.5)	5.79K $\pm$ 81	71.63K $\pm$ 2.04K	0.710 $\pm$ 0.007	0.524 $\pm$ 0.008	17.81 $\pm$ 0.33

4.2. Model Configurations

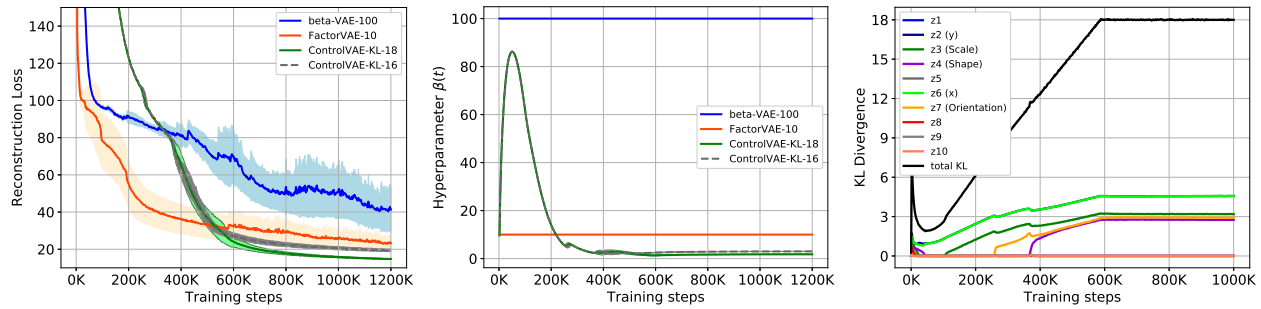
The detailed model configurations and hyperparameter settings for each model is presented in Appendix A.

4.3. Evaluation on Language Modeling

First, we compare the performance of ControlVAE with the following baselines for mitigating KL vanishing in text generation (Bowman et al., 2015).
Cost annealing (Bowman et al., 2015): This method gradually increases the hyperparameter on KL-divergence from 0 until to 1 after N training steps using sigmoid function.
Cyclical annealing (Liu et al., 2019): This method splits the training process into M cycles and each increases the hyperparameter from 0 until to 1 using a linear function.

Fig. 3 illustrates the comparison results of KL-divergence, reconstruction loss and hyperparamter $\beta(t)$ for different methods on the PTB dataset. Note that, here ControlVAE-KL-v means we set the KL-divergence to a desired value v (e.g., 3) for our PI controller following the set point guidelines in Section 3.2. Cost-annealing-v means we gradually increase the hyperparameter, $\beta(t)$, from 0 until to 1 after v steps using sigmoid function. We observe from Fig. 3(a) that ControlVAE (KL=1.5, 3) and Cyclical annealing $(4, 8$ cycles) can avert the KL vanishing. However, our ControlVAE is able to stabilize the KL-divergence while cyclical annealing could not. Moreover, our method has a lower reconstruction loss than the cyclical annealing in Fig. 3 (b). Cost annealing method still suffers from KL vanishing, because we use the Transformer (Vaswani et al., 2017) as the decoder, which can predict the current data based on previous ground-truth data. Fig. 3 (c) illustrates the tuning result of $\beta(t)$ by ControlVAE compared with other methods. We can discover that our $\beta(t)$ gradually converges to around a certain value. Note that, here $\beta(t)$ of ControlVAE does not converge to 1 because we slightly increase the value of KL-divergence (produced by the original VAE) in order to improve the diversity of generated data.

In order to further demonstrate ControlVAE can improve the diversity of generated text, we apply it to dialog-response generation using the Switchboard(SW) dataset. Following (Zhao et al., 2017b), we adopt a conditional VAE (Zhao et al., 2017b) that generates dialog conditioned on the previous response. We use metric Dis-n (Xu et al., 2018) and self-BLEU (Zhu et al., 2018) (with 1000 sampled results) to measure the diversity of generated data, and perplexity (PPL) (Jelinek et al., 1977) to measure how well the probability distribution predicts a sample. Table 1 illustrates the comparison results for different approaches. We can observe that ControlVAE has more distinct grams and lower self-BLEU than the baselines when the desired KL-divergence is set to 35 and 25. In addition, it has lower PPL than the other methods. Thus, we can conclude that ControlVAE can improve the diversity of generated data and generation performance. We also illustrate some examples of generated dialog by ControlVAE in Appendix B.



(a) Reconstruction loss (b) $\beta(t)$ (c) Disentangled factors

Figure 4. (a) (b) shows the comparison of reconstruction error and $\beta(t)$ using 2D Shapes data over 5 random seeds. ControlVAE (KL=16, 18) has lower reconstruction errors and variance compared to β-VAE. (c) shows an example about the disentangled factors in the latent variable as the total KL-divergence increases from 0.5 to 18 for ControlVAE (KL=18). Each curve with positive KL-divergence (except black one) represents one disentangled factor by ControlVAE.

Table 2. Performance comparison of different methods using disentanglement metric, MIG score, averaged over 5 random seeds. The higher is better. ControlVAE (KL=16) has a comparable MIG score but lower variance than the FactorVAE with the default parameters.

Metric	ControlVAE (KL=16)	ControlVAE (KL=18)	β-VAE ($\beta = 100$)	FactorVAE ($\gamma = 10$)
MIG	**0.5628 ± 0.0222**	0.5432 ± 0.0281	0.5138 ± 0.0371	0.5625 ± 0.0443

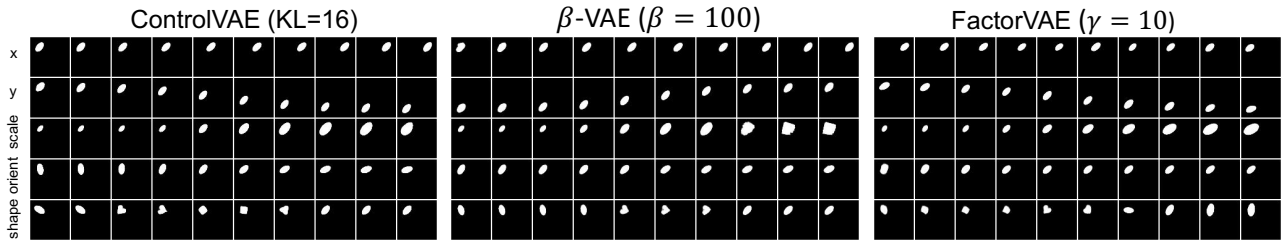


Figure 5. Rows: latent traversals ordered by the value of KL-divergence with the prior in a descending order. Following work (Burgess et al., 2018), we initialize the latent representation from a seed image, and then traverse a single latent dimension in a range of $[-3, 3]$, while keeping the remaining latent dimensions fixed. ControlVAE can disentangle all the five generative factors for 2D Shapes data, while β-VAE entangles the scale and shape (in 3rd row) and FactorVAE does not disentangle orientation (in 4th row) very well.

4.4. Evaluation on Disentangled Representations

We then evaluate the performance of ControlVAE on the learning of disentangled representations using *2D Shapes* data. We compare it with two baselines: FactorVAE (Kim & Mnih, 2018) and β-VAE (Burgess et al., 2018).

Fig. 4 (a) and (b) shows the comparison of reconstruction error and the hyperparameter $\beta(t)$ (using 5 random seeds) for different models. We can observe from Fig. 4 (a) that ControlVAE (KL=16,18) has lower reconstruction error and variance than the baselines. This is because our ControlVAE automatically adjusts the hyperparameter, $\beta(t)$, to stabilize the KL-divergence, while the other two methods keep the hyperparameter unchanged during model training. Specifically, for ControlVAE (KL=18), the hyperparameter $\beta(t)$ is high in the beginning in order to obtain good disentangling, and then it gradually drops to around 1.8 as the training converges, as shown in Fig. 4(b). In contrast, β-VAE ($\beta = 100$) has a large and fixed weight on the KL-divergence so that its optimization algorithm tends to optimize the KL-divergence term, leading to a large reconstruction error. In addition, Fig. 4(c) illustrates an example of KL-divergence per factor in the latent code as training progresses and the total information capacity (KL-divergence) increases from 0.5 until to 18. We can see that ControlVAE disentangles all the five generative factors, starting from positional latents (x and y) to scale, followed by orientation and then shape.

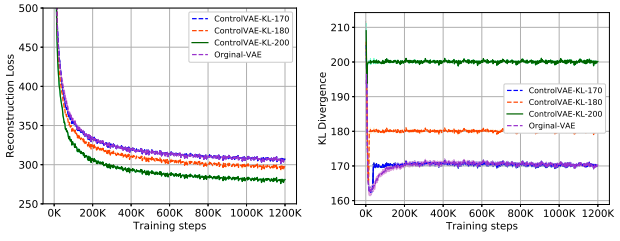
To further demonstrate ControlVAE can achieve a better disentangling, we use a disentanglement metric, mutual information gap (MIG) (Chen et al., 2018a), to compare their performance, as shown in Table 2. It can be observed that ControlVAE (KL=16) has a comparable MIG but lower variance than FactorVAE. Here it is worth noting that Fac-

torVAE adds a Total Correlation (TC) term in the objective while our method does not. Since there does not exist a robust metric to fully measure disentanglement, we also show the qualitative results of different models in Fig. 5. We can observe that ControlVAE can discover all the five generative factors: positional latent (x and y), scale, orientation and shape. However, β-VAE ($\beta = 100$) disentangles four generative factors except for entangling the scale and shape together (in the third row), while FactorVAE ($\gamma = 10$) does not disentangle the orientation factor very well in the fourth row in Fig. 5. Thus, ControlVAE achieves a better reconstruction quality and disentangling than the baselines.

4.5. Evaluation on Image Generation

Finally, we compare the reconstruction quality on image generation task for ControlVAE and the original VAE. Fig. 6 shows the comparison of reconstruction error and KL-divergence under different desired values of KL-divergence averaged over 3 random seeds. We can see from Fig. 6(a) that ControlVAE-KL-200 (KL=200) has the lowest reconstruction error among them. This is because there exists a trade-off between the reconstruction accuracy and KL-divergence in the VAE objective. When the value of KL-divergence rises, which means the weight of KL term decreases, the model tends to optimize the reconstruction term. In addition, as we set the desired KL-divergence to 170 (same as the basic VAE in Fig. 6(b)), ControlVAE has the same reconstruction error as the original VAE. At that point, ControlVAE becomes the original VAE as $\beta(t)$ finally converges to 1, as shown in Fig. 7 in Appendix C.

We further adopt two commonly used metrics for image generation, FID (Lucic et al., 2018) and SSIM (Chen et al., 2018b), to evaluate the performance of ControlVAE in Table 3. It can be observed that ControlVAE-KL-200 outperforms the other methods in terms of FID and SSIM. Therefore, ControlVAE can improve the reconstruction quality for image generation task via slightly increasing the value of KL-divergence. We also show some examples to verify ControlVAE has a better reconstruction quality than the basic VAE in Appendix D.



(a) Reconstruction loss (b) KL-divergence

Figure 6. Performance comparison for different methods on the CelebA data.

Table 3. Performance comparison for different methods on CelebA data over 3 random seeds. FID: lower is better. SSIM: higher is better.

Methods/metric	FID	SSIM
ControlVAE-KL-200	**55.16** $\pm$ 0.187	**0.687** $\pm$ 0.0002
ControlVAE-KL-180	57.57 $\pm$ 0.236	0.679 $\pm$ 0.0003
ControlVAE-KL-170	58.75 $\pm$ 0.286	0.675 $\pm$ 0.0001
Original VAE	58.71 $\pm$ 0.207	0.675 $\pm$ 0.0001

5. Related Work

We review the related work about VAE and its variants in various applications, and then point out the difference between our method and the prior work.

There are many work involving a trade-off between reconstruction and KL-divergence for VAEs applications. For disentangled representation learning, researchers proposed β-VAE ($\beta > 1$) (Higgins et al., 2017; Burgess et al., 2018) that assigns a large and fixed hyperparameter, β, to put more emphasis on the KL divergence to encourage disentangled latent representations. It, however, sacrifice the reconstruction quality in order to obtain better disentangling. Then some follow-up work (Chen et al., 2018a; Kim & Mnih, 2018) further factorize the KL-divergence term to improve the reconstruction quality. However, these methods still assign a fixed and large hyperparameter to the decomposed terms in the objective, resulting in high reconstruction error. In contrast, ControlVAE dynamically tunes β during optimization to achieve good disentangling *and* better reconstruction quality. More importantly, our PI control method can be used as a plug-in replacement of existing methods.

In order to improve the sample generation quality of VAEs (Dai & Wipf, 2019; Zhao et al., 2019; Xiao et al., 2019; Ghosh et al., 2019; Alemi et al., 2017; Zhao et al., 2019), some researchers tried to reduce the weight of KL-divergence to make the decoder produce sharper outputs. Though they can obtain impressive sample quality, they suffer severely from the trade-off in the way that the latent distribution is far away from the prior. Recent studies adopted a constrained optimization for reconstruction error (Rezende & Viola, 2018; Klushyn et al., 2019) to achieve the trade-off between reconstruction error and KL-divergence. However, they may suffer from posterior collapse if the inference network fails to cover the latent space while our can totally avert posterior collapse. In addition, different from their work, we try to optimize KL-divergence (information bottleneck) as a constraint. Our method and theirs complement each other for different applications.

In language modeling, VAE often suffers from KL vanishing, due to a powerful decoder, such as Transformer (Vaswani et al., 2017) and LSTM. To remedy this

issue, one popular way is to add a hyperparameter β on the KL term (Bowman et al., 2015; Liu et al., 2019), and then gradually increases it from 0 until 1. However, the existing methods (Yang et al., 2017; Bowman et al., 2015; Liu et al., 2019), such as KL cost annealing and cyclical annealing, cannot totally solve KL vanishing or explicitly control the value of KL-divergence since they blindly change β without observing the actual KL-divergence during model training. Conversely, our approach can avert KL vanishing and stabilize the KL-divergence to a desired value.

6. Conclusion and Future Work

In this paper, we proposed a general controllable VAE framework, ControlVAE, that combines automatic control with the basic VAE framework to improve the performance of the VAE models. We designed a new non-linear PI controller to control the value of KL divergence during model training. Then we evaluated ControlVAE on three different tasks. The results show that ControlVAE attains better performance; it can achieve a higher reconstruction quality and good disentanglement. It can totally avert KL vanishing and improve the diversity of generated data in language modeling. In addition, it improves the reconstruction quality on the task of image generation. For future work, we plan to apply our method to other research topics, such as topic modeling and semi-supervised applications.

Acknowledgments

Research reported in this paper was sponsored in part by DARPA award W911NF-17-C-0099, DTRA award HDTRA1-18-1-0026, and the Army Research Laboratory under Cooperative Agreements W911NF-09-2-0053 and W911NF-17-2-0196.

References

Alemi, A. A., Poole, B., Fischer, I., Dillon, J. V., Saurous, R. A., and Murphy, K. Fixing a broken elbo. *arXiv preprint arXiv:1711.00464*, 2017.

Åström, K. J., Hägglund, T., and Astrom, K. J. *Advanced PID control*, volume 461. ISA-The Instrumentation, Systems, and Automation Society Research Triangle , 2006.

Azar, A. T. and Serrano, F. E. Design and modeling of anti wind up pid controllers. In *Complex system modelling and control through intelligent soft computations*, pp. 1–44. Springer, 2015.

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

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

Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., and Lerchner, A. Understanding disentangling in beta-vae. *arXiv preprint arXiv:1804.03599*, 2018.

Chen, T. Q., Li, X., Grosse, R. B., and Duvenaud, D. K. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, pp. 2610–2620, 2018a.

Chen, X., Xu, C., Yang, X., and Tao, D. Attention-gan for object transfiguration in wild images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 164–180, 2018b.

Dai, B. and Wipf, D. Diagnosing and enhancing vae models. *arXiv preprint arXiv:1903.05789*, 2019.

Denton, E. L. et al. Unsupervised learning of disentangled representations from video. In *Advances in neural information processing systems*, pp. 4414–4423, 2017.

Ghosh, P., Sajjadi, M. S., Vergari, A., Black, M., and Schölkopf, B. From variational to deterministic autoencoders. *arXiv preprint arXiv:1903.12436*, 2019.

Godfrey, J. and Holliman, E. Switchboard-1 release 2-linguistic data consortium. *SWITCHBOARD: A Users Manual*, 1997.

Hellerstein, J. L., Diao, Y., Parekh, S., and Tilbury, D. M. *Feedback control of computing systems*. John Wiley & Sons, 2004.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2(5):6, 2017.

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

Hu, Z., Shi, H., Tan, B., Wang, W., Yang, Z., Zhao, T., He, J., Qin, L., Wang, D., et al. Texar: A modularized, versatile, and extensible toolkit for text generation. In *ACL 2019, System Demonstrations*, 2019.

Jelinek, F., Mercer, R. L., Bahl, L. R., and Baker, J. K. Perplexitya measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63, 1977.

Kim, H. and Mnih, A. Disentangling by factorising. In *International Conference on Machine Learning*, pp. 2654–2663, 2018.

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

Klushyn, A., Chen, N., Kurle, R., Cseke, B., and van der Smagt, P. Learning hierarchical priors in vaes. In *Advances in Neural Information Processing Systems*, pp. 2866–2875, 2019.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Liu, M.-Y., Breuel, T., and Kautz, J. Unsupervised image-to-image translation networks. In *Advances in neural information processing systems*, pp. 700–708, 2017.

Liu, X., Gao, J., Celikyilmaz, A., Carin, L., et al. Cyclical annealing schedule: A simple approach to mitigating kl vanishing. *arXiv preprint arXiv:1903.10145*, 2019.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015.

Lucic, M., Kurach, K., Michalski, M., Gelly, S., and Bousquet, O. Are gans created equal? a large-scale study. In *Advances in neural information processing systems*, pp. 700–709, 2018.

Marcus, M., Santorini, B., and Marcinkiewicz, M. A. Building a large annotated corpus of english: The penn treebank. 1993.

Matthey, L., Higgins, I., Hassabis, D., and Lerchner, A. dsprites: Disentanglement testing sprites dataset. *URL https://github. com/deepmind/dsprites-dataset/.[Accessed on: 2018-05-08]*, 2017.

Peng, Y., Vrancic, D., and Hanus, R. Anti-windup, bumpless, and conditioned transfer techniques for pid controllers. *IEEE Control systems magazine*, 16(4):48–57, 1996.

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

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

Sohn, K., Lee, H., and Yan, X. Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems*, pp. 3483–3491, 2015.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Wang, W., Gan, Z., Xu, H., Zhang, R., Wang, G., Shen, D., Chen, C., and Carin, L. Topic-guided variational autoencoders for text generation. *arXiv preprint arXiv:1903.07137*, 2019.

Xiao, Z., Yan, Q., Chen, Y., and Amit, Y. Generative latent flow: A framework for non-adversarial image generation. *arXiv preprint arXiv:1905.10485*, 2019.

Xu, J., Ren, X., Lin, J., and Sun, X. Dp-gan: Diversity-promoting generative adversarial network for generating informative and diversified text. *arXiv preprint arXiv:1802.01345*, 2018.

Yan, X., Yang, J., Sohn, K., and Lee, H. Attribute2image: Conditional image generation from visual attributes. In *European Conference on Computer Vision*, pp. 776–791. Springer, 2016.

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

Zhao, S., Song, J., and Ermon, S. Towards deeper understanding of variational autoencoding models. *arXiv preprint arXiv:1702.08658*, 2017a.

Zhao, S., Song, J., and Ermon, S. Infovae: Balancing learning and inference in variational autoencoders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 5885–5892, 2019.

Zhao, T., Zhao, R., and Eskenazi, M. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. *arXiv preprint arXiv:1703.10960*, 2017b.

Zhu, Y., Lu, S., Zheng, L., Guo, J., Zhang, W., Wang, J., and Yu, Y. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 1097–1100, 2018.

A. Model Configurations and hyperparameter settings

We summarize the detailed model configurations and hyperparameter settings for ControlVAE in the following three applications: language modeling, disentanglement representation learning and image generation.

A.1. Experimental Details for Language Modeling

For text generation on PTB data, we build the ControlVAE model on the basic VAE model, as in (Bowman et al., 2015). We use one-layer LSTM as the encoder and a three-layer Transformer with eight heads as the decoder and a Multi-Layer Perceptron (MLP) to learn the latent variable $\mathbf{z}$. The maximum sequence length for LSTM and Transformer is set to 100, respectively. And the size of latent variable is set to 64. Then we set the dimension of word embedding to 256 and the batch size to 32. In addition, the dropout is 0.2 for LSTM and Transformer. Adam optimization with the learning rate 0.001 is used during training. Following the tuning guidelines above, we set the coefficients K_p and K_i of P term and I term to 0.01 and 0.0001, respectively. Finally, We adopt the source code on Texar platform to implement experiments (Hu et al., 2019).

For dialog-response generation, we follow the model architecture and hyperparameters of the basic conditional VAE in (Zhao et al., 2017b). We use one-layer Bi-directional GRU as the encoder and one-layer GRU as the decoder and two fully-connected layers to learn the latent variable. In the experiment, the size of both latent variable and word embeddings is set to 200. The maximum length of input/output sequence for GRU is set to 40 with batch size 30. In addition, Adam with initial learning rate 0.001 is used. In addition, we set the same K_p and K_i of PI algorithm as text generation above. The model architectures of ControlVAE for these two NLP tasks are illustrated in Table 4, 5.

Table 4. Encoder and decoder architecture for text generation on PTB data.

Encoder	Decoder
Input n words $\times 256$	Input $\in \mathbb{R}^{64}$, $n \times 256$
1-layer LSTM	FC 64×256
FC 64×2	3-layer Transformer 8 heads

A.2. Experimental Details for Disentangling

Following the same model architecture of β-VAE (Higgins et al., 2017), we adopt a convolutional layer and deconvolutional layer for our experiments. We use Adam optimizer with $\beta_1 = 0.90$, $\beta_2 = 0.99$ and a learning rate tuned from 10^{-4}. We set K_p and K_i for PI algorithm to 0.01 and 0.001, respectively. For the step function, we set the step, α, to 0.15 per $K = 5000$ training steps as the information capacity (desired KL- divergence) increases from 0.5 until 18 for 2D Shape data. ControlVAE uses the same encoder and decoder architecture as β-VAE except for plugging in PI control algorithm, illustrated in Table 6.

Table 5. Encoder and decoder architecture for dialog generation on Switchboard (SW) data.

Encoder	Decoder
Input n words $\times 200$	Input $\in \mathbb{R}^{200}$
1-layer bi-GRU	FC 200×400
FC 200×2	1-layer GRU
FC 200×2	

Table 6. Encoder and decoder architecture for disentangled representation learning on 2D Shapes data.

Encoder	Decoder
Input 64×64 binary image	Input $\in \mathbb{R}^{10}$
4×4 conv. 32 ReLU. stride 2	FC. 256 ReLU.
4×4 conv. 32 ReLU. stride 2	4×4 upconv. 256 ReLU.
4×4 conv. 64 ReLU. stride 2	4×4 upconv. 64 ReLU. stride 2.
4×4 conv. 64 ReLU. stride 2	4×4 upconv. 64 ReLU. stride 2
4×4 conv. 256 ReLU.	4×4 upconv. 32 ReLU. stride 2
FC 256. FC. 2×10	4×4 upconv. 32 ReLU. stride 2

A.3. Experimental Details for Image Generation

Similar to the architecture of β-VAE, we use a convolutional layer with batch normalization as the encoder and a deconvolutional layer with batch normalization for our experiments. We use Adam optimizer with $\beta_1 = 0.90$, $\beta_2 = 0.99$ and a learning rate 10^{-4} for CelebA data. The size of latent variable is set to 500, because we find it has a better reconstruction quality than 200 and 400. In addition, we set the desired value of KL-divergence to 170 (same as the original VAE), 180, and 200. For PI control algorithm, we set K_p and K_i to 0.01 and 0.0001, respectively. We also use the same encoder and decoder architecture as β-VAE above except that we add the batch normalization to improve the stability of model training, as shown in Table 7.

B. Examples of Generated Dialog by ControlVAE

In this section, we show an example to compare the diversity and relevance of generated dialog by different methods, as illustrated in Table 8. Alice begins with the open-ended

Table 7. Encoder and decoder architecture for image generation on CelebA data.

Encoder	Decoder
Input $128 \times 128 \times 3$ RGB image	Input $\in \mathbb{R}^{500}$
4×4 conv. 32 ReLU. stride 2	FC. 256 ReLU.
4×4 conv. 32 ReLU. stride 2	4×4 upconv. 256 ReLU. stride 2
4×4 conv. 64 ReLU. stride 2	4×4 upconv. 64 ReLU. stride 2.
4×4 conv. 64 ReLU. stride 2	4×4 upconv. 64 ReLU. stride 2
4×4 conv. 256 ReLU. stride 2	4×4 upconv. 32 ReLU. stride 2
FC 4096. FC.2 $\times$ 500	4×4 upconv. 32 ReLU. stride 2

conversation on choosing a college. Our model tries to predict the response from Bob. The ground truth response is "um - hum". We can observe from Table 8 that ControlVAE (KL=25, 35) can generate diverse and relevant response compared with the ground truth. In addition, while cyclical annealing can generate diverse text, some of them are not very relevant to the ground-truth response.

Table 8. Examples of generated dialog for different methods. Our model tries to predict the response from Bob. The response generated by ControlVAE (KL=25,35) are relevant and diverse compared with the ground truth. However, some of reponse generated by cost annealing and cyclical annealing are not very relevant to the ground-truth data

Context: (Alice) and a lot of the students in that home town sometimes ⟨ unk ⟩ the idea of staying and going to school across the street so to speak **Topic**: Choosing a college **Target**: (Bob) um - hum	
ControlVAE-KL-25	**ControlVAE-KL-35**
yeah	uh - huh
um - hum	yeah
oh that's right um - hum	oh yeah oh absolutely
yes	right
right	um - hum
Cost annealing (KL=17)	**Cyclical anneal (KL=21.5)**
oh yeah	yeah that's true do you do you do it
uh - huh	yeah
right	um - hum
uh - huh and i think we have to be together	yeah that's a good idea
oh well that's neat yeah well	yeah i see it too,it's a neat place

C. $\beta(t)$ of ControlVAE for Image Generation on CelebA data

Fig. 7 illustrates the comparison of $\beta(t)$ for different methods during model training. We can observe that $\beta(t)$ finally converges to 1 when the desired value of KL-divergence is set to 170, same as the original VAE. At this point, ControlVAE becomes the original VAE. Thus, ControlVAE can be customized by users based on different applications.

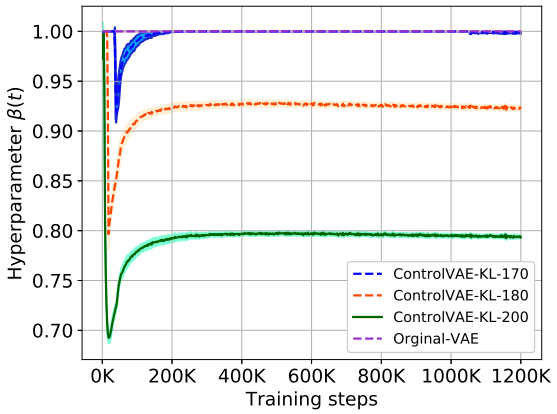
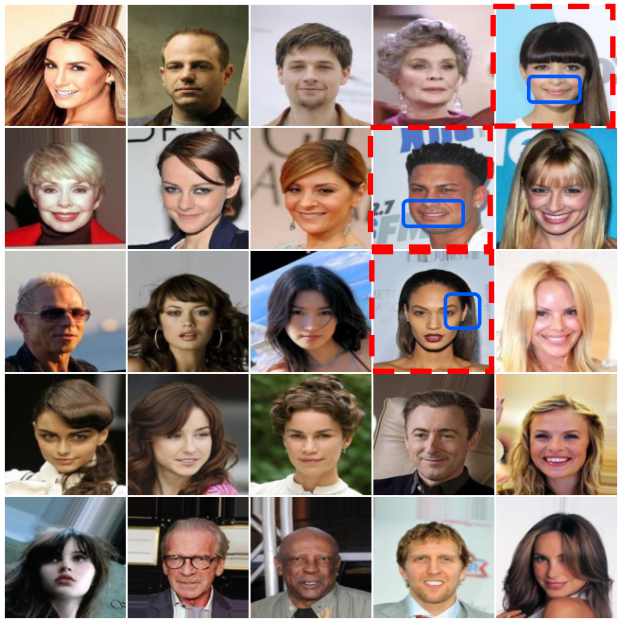


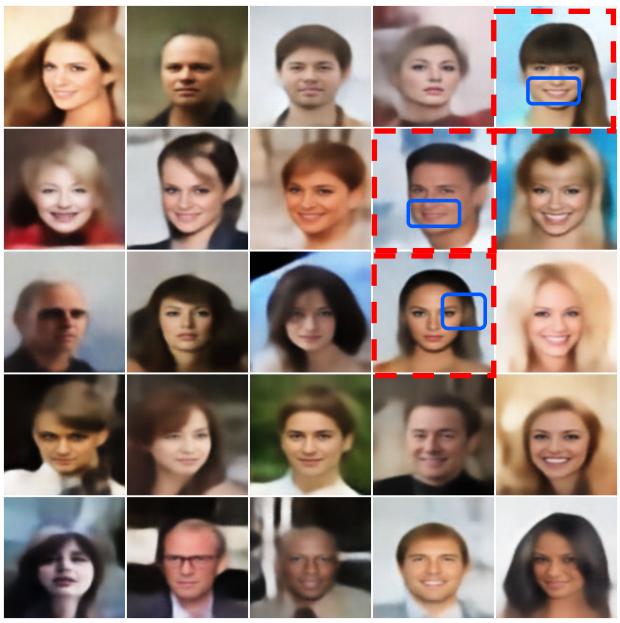
Figure 7. Hyperparameter $\beta(t)$ of ControlVAE for image generation on CelebA data for 3 random seeds. If we set the desired value of KL-divergence to 170, the hyperparameter, $\beta(t)$, gradually approaches 1. It means the ControlVAE becomes the original VAE.

D. Examples of Reconstruction Images by VAE and ControlVAE

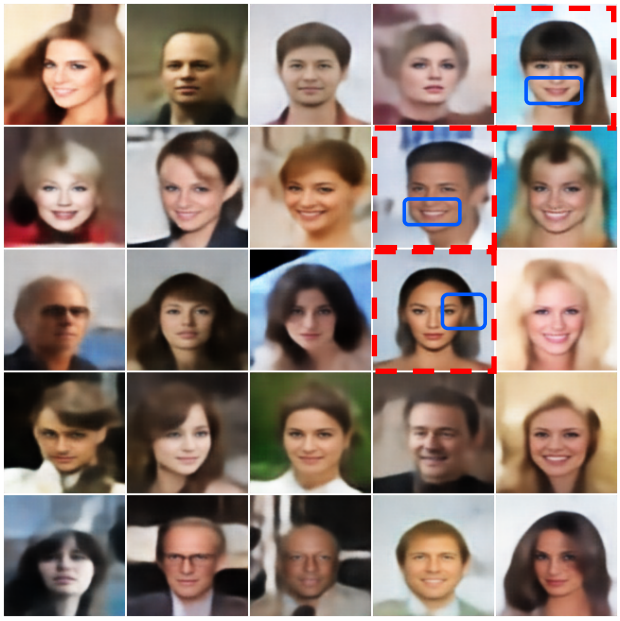
We also show some reconstruction images by ControlVAE and the original VAE in Fig. 8. It can be observed that images reconstructed by ControlVAE-KL-200 (KL = 200) has the best reconstruction quality compared to the original VAE. Take the woman in the first row last column as an example. The woman does not show her teeth in the ground-truth image. However, we can see the woman reconstructed by the original VAE smiles with mouth opening. In contrast, the woman reconstructed by ControlVAE-KL-200 hardly show her teeth when smiling. In addition, we also discover from the other two examples marked with blue rectangles that ControlVAE-KL-200 can better reconstruct the "smile" from the man and the "ear" from the woman compared to the original VAE. Therefore, we can conclude that our ControlVAE can improve the reconstruction quality via slightly increasing (control) KL-divergence compared to the original VAE. Note that, if we want to improve the generation quality by sampling the latent code, we can reduce the KL divergence, which will be explored in the future.



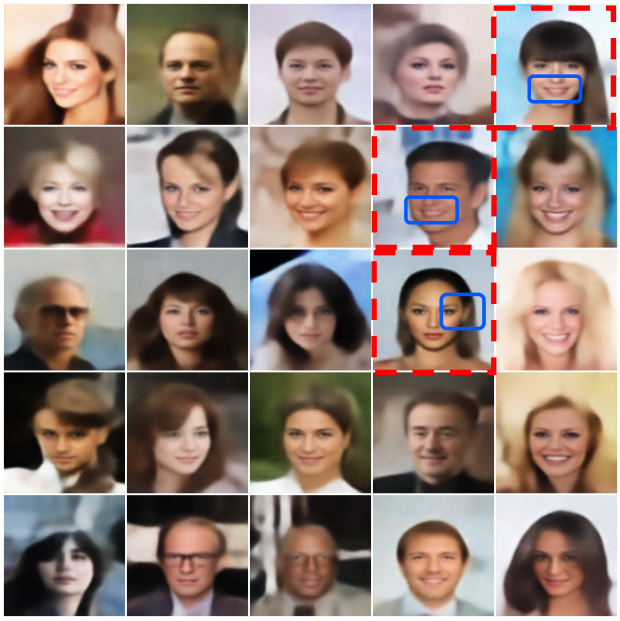
(a) Ground truth



(b) Original VAE



(c) ControlVAE-KL-200



(d) ControlVAE-KL-170

Figure 8. Examples of images recontructed by different methods and ground truth. From the images marked with blue rectangles, we can see that ControlVAE-KL-200 (KL=200) can better reconstruct woman's month opening (first row last column), man's smiling with teeth (second row fourth column), and woman'ear (third row fourth column) than the original VAE based on the ground-truth data in (a).