On the Anatomy of Latent-variable Generative Models for Conditional Text Generation

Anonymous ACL submission

Abstract

Conditional text generation is a non-trivial task, which is until now predominantly performed with latent-variable generative models. In this work, we intend to explore several choices that are shown to affect the two essential aspects of model performance: expressivity and controllability. We propose to experiment with a series of latent-variable models built around simple design changes under a general unified framework, with a particular focus on prior distributions based on Energy-Based Models instead of the usual standard Gaussian. Our experiments validate the claim that this richer prior allows for a better representational power, but it exhibits difficult training. We provide a comprehensive analysis of these difficulties and a close comparison with recent work on EBM-based priors for conditional text generation[1].

1 Introduction

Conditional (or controllable) text generation consists in generating realistic textual language while controlling an attribute variable. There is a large variety of attributes that one could condition content generation on, depending on the application: we can mention dialog models, with control over intent in the conversation (Zhao et al., 2017), or story generation, with control over the persona (Chandu et al., 2019), among others. In controllable text generation, attributes are commonly encoded as control vectors (Prabhumoye et al., 2020). In this setting, it is natural to use generative models based on latent representations (Bowman et al., 2016; Kim et al., 2018; Pelsmaeker and Aziz, 2020). Instead of estimating directly the data distribution in the observation space, latent-variable generative models define a continuous latent variable and learn its distribution. Then, text can be generated by sampling a prior in the case of Variational Autoencoder (VAEs) (Kingma and Welling, 2014; Rezende et al., 2014). Adapting these approaches to conditional generation can be achieved by integrating an additional latent *attribute* variable into the model.

In conditional text generation, model performance can be assessed against two properties: i) the quality of the generated text, which should be realistic, but also diverse and ii), the ability of the approach to effectively generate content that corresponds to the attribute value. Firstly, the quality of the generated text depends greatly on the latent-variable model used. On that matter, most models (Bowman et al., 2016; Yang et al., 2017; Kim et al., 2018) use a simple prior distribution like the standard Gaussian, or the uniform distribution. There is a large number of works investigating more expressive priors, with several of them focusing on text, leading to improved results on text modeling tasks (Zhao et al., 2018; Ding and Gimpel, 2021). Secondly, controllability offers a particular challenge, as continuous latent variables are not naturally adapted to represent discrete attributes. Several ideas (Hu et al., 2017; Li et al., 2020; Duan et al., 2020) have been proposed to facilitate controlled generation: however, those applied to latent-variable generative models usually need to use supplementary classifiers or generators, which requires many more parameters, or to optimizing simultaneously an adversarial objective or a regularization term, which is difficult and may lead to poor control abilities.

In this paper, we are interested in exploring how simple factors in model design affect those two aspects of model performance, away from more elaborate solutions recently proposed in the literature. To achieve this, we propose a framework based on a latent-variable generative model, in which we propose to vary (1) the complexity of the prior; (2) the way the attribute and latent representation interact; (3) the learning procedure. We thus aim at providing a clear view of the impact and usefulness of

[1] We will release our code upon publication to facilitate future work.

each design choice on conditional text generation tasks. We choose to make our framework generalize a very recent work (Pang and Wu, 2021) investigating conditional text generation with a model that learns the prior distribution of the latent space with an Energy-Based Model (EBM) (Fahlman et al., 1983; Smolensky, 1986; Zhu et al., 1998; Salakhutdinov and Hinton, 2009; Rosenfeld et al., 2001; Wang et al., 2015; Lu et al., 2016; Wang and Ou, 2017), which was previously explored by Pang et al. (2020) for unconditional generation. We will first explore the related literature and motivate our design choices in Section 2; then, state our problem and detail our framework in Sections 3 and 4. In Section 5, we check the performance of our models experimentally by measuring the quality of the generated text and evaluating how well it is able to control the attribute of sentences through the accuracy of an external classifier. Finally, we discuss the particular issues raised by the training of an EBM prior and expand on the comparison with the work of Pang and Wu (2021) in Section 6. To summarize, our contributions are as follows:

1. Taking a step back from the often complex literature on the subject, we provide a clear view of how several factors impact the performance of latent-variable models for conditional text generation.

2. We experiment with various models within our framework on two datasets; in particular, we provide a comprehensive study of the EBM-based prior and draw a fine-grained comparison with a recently published work also employing this prior for the same task (Pang and Wu, 2021).

3. We find out that while it has a better representational power, an EBM-based prior is very difficult to train, and that our best performing model is akin to the S-VAE (Kingma et al., 2014).

2 Related Works

2.1 Expressivity in latent-variable generative models

Among latent-variable models, VAEs are often thought of as having their expressivity limited by simplistic priors – they usually employ simple gaussian distributions – and by the restrictive assumption it puts on their latent posterior (Ding and Gimpel, 2021). Researchers have tried to improve the representational expressivity of their models through the use of more complex priors, such as mixture of gaussians (Wang et al., 2019), the Dirichlet distribution (Burkhardt and Kramer, 2019) or the Variational Mixture of Posteriors Prior (VampPrior) (Tomczak and Welling, 2018), but also recently with priors based on on normalizing flows (Ding and Gimpel, 2021). Another way to do so is to directly learn a parametrized model as prior: the Variational Lossy Autoencoder (Chen et al., 2017) parameterizes the prior with a learnable autoregressive flow from a simple gaussian distribution, while ARAE (Zhao et al., 2018) learns the prior through a generator model with adversarial learning. This is also the idea behind the EBM-based prior of Pang et al. (2020): interestingly, the authors do not use variational inference, but resort to sampling for exact posterior inference with Markov Chain Monte Carlo (MCMC). Thus, they are avoiding any assumption about the form taken by the posterior distribution, which is also the motivation behind the work of Fang et al. (2019): they propose to learn implicitly the posterior, to avoid it being gaussian-based. In this work, we propose to investigate the use of a flexible EBM-based prior, and to compare it to the usual gaussian prior. However, given the assumptions accompanying both gaussian priors and variational inference, we believe we should not make such a change without also investigating how it interacts with the learning process. Hence, we train our models with both Expectation-Maximization (EM), and Variational Inference (VI).

2.2 Controllability in conditional text generation

Most of the existing approaches to conditional text generation are based on latent-variable models: however, they vary greatly in how they deal with attribute information. Some integrate the attribute into the latent space; for example Shi et al. (2020) uses a gaussian mixture prior, where to each component corresponds an attribute class. They add a dispersion term to the training objective to avoid mode collapse and force latent representations corresponding to different attributes into well-separated clusters. Contrarily, attributes may come from an external source. Then, models differ in how they make the attribute information interact with the latent representation: Hu et al. (2017); Li et al. (2020) are focused on disentangling the attribute information from the rest of the representation, using an auxiliary classifier that discriminates between the generated examples matching the attribute and those that do not. For each possible attribute, Duan et al. (2020) map the latent space of a pre-trained VAE into a smaller attribute-exclusive space with an individual *plugin* VAE, which has the advantage of allowing for semi-supervised learning, as attribute information is only needed for training these plugins. Recent approaches based on large language models also fit in this second category: similarly to Li et al. (2020), Keskar et al. (2019) use control codes as a separate input to the model (which implies training it from scratch), while Dathathri et al. (2020) uses gradient information from a classifier trained on the desired attributes to explore the hidden space of a pre-trained model. In this work, we propose avoid any complicated solution and to only make a simple change to how an external attribute variable and the latent representation interact, by making them independent, or conditionally independent given the observation, and compare the behaviour of both approaches.

2.3 EBMs for text generation

Energy-based models have often been used for sequence modeling (Wang et al., 2015; Wang and Ou, 2017), with a recent growth in popularity: with autoregressive generative models, for calibration (He et al., 2021), efficient scoring (Clark et al., 2020); but also for non-autoregressive general purpose text generation (Deng et al., 2020), or in machine translation (Tu et al., 2020). However, the discrete space of textual data implies using methods like Noise-Contrastive Estimation (Gutmann and Hyvärinen, 2010). Pang et al. (2020) moves the energy modeling into a continuous latent space, of much lower dimension, making it easier to apply the model to textual data. The closest existing approach to our work, the Symbol-Vector Coupling Energy-Based Model (SVEBM) of Pang and Wu (2021), uses an inference network to approximate the intractable posterior distribution of the latent variable, and regularization based on the information bottleneck to ensure the latent representation contains information from the controlling attribute. However, the attribute directly intervenes in the EBM, which actually models both the attribute and the prior jointly. In this paper, we adopt a wider approach and propose an EBM prior separated from the attribute. We also carry out a thorough comparison of our framework with the SVEBM of Pang and Wu (2021).

3 Problem and notations

All along this paper, we represent a text sequence by a random sequence X over a vocabulary $\mathcal{V}$. In general, an observed text sequence of size L is a realization of X denoted by $x = (x^t)_{l=1}^L$, where each word/token x^t belongs to $\mathcal{V}$. In this paper, the attribute Y is a categorical variable taking its values in the set $\mathcal{Y} = \{1, \dots, m\}$. Generating text conditioned on an attribute can be seen as drawing a family of conditional distribution $(\mathrm{P}_{X|y})_{y\in\mathcal{Y}}$. We assume to observe pairs $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ for $i = 1, \dots n$ where y_i is the attribute value on which the generation of the text sequence x_i has been conditioned. In this setting, our goal is to learn a parametric model of the family of conditional distributions $(\mathrm{P}_{X|y})_{y\in\mathcal{Y}}$ from a set of n observations $D_n = \{(x_i, y_i)_{i=1}^n\}$.

4 Latent-variable Generative Model for Conditional Generation

To address the learning problem described above, latent-variable generative models seek to obtain an internal representation that explains the observation x, through a random latent variable Z. In this work, we restrict ourselves to the case where Z is continuous, taking its values in $\mathbb{R}^d$, and we define a probabilistic graphical parametric model P_θ to estimate the joint distribution of observed data variable X, condition variable Y and latent unstructured variable Z. $p_\theta(x, y, z)$ is the density of the model, and can be factorized as $p_\alpha(z) \times p_\beta(x, y|z)$. The following section is dedicated to the definition of the models we will study within this framework, varying with respect to (1) how to model the latent prior $p_\alpha(z)$, (2) the further factorization of $p_\beta(x, y|z)$ and (3) the learning procedure.

4.1 Latent prior $p_\alpha(z)$

The usual choice for the latent distribution $p_\alpha(z)$ is a standard Gaussian $\mathcal{N}(0, \mathbf{I})$, following the VAE (Kingma and Welling, 2014) and S-VAE (Kingma et al., 2014) models; in that case, the parameter of the distribution α is fixed beforehand. In Pang et al. (2020); Pang and Wu (2021), the density of the EBM serving as prior for the latent space $\mathcal{Z}$ is defined as follows:

$$p_\alpha(z) = \frac{1}{C(\alpha)} \exp(f_\alpha(z)) \times \mathcal{N}(z; 0, \mathbf{I}) \quad (1)$$

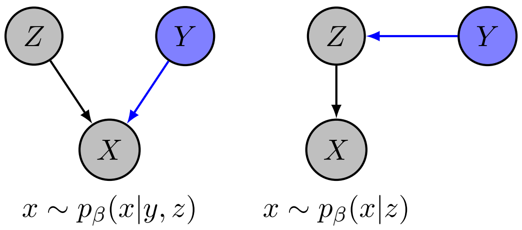

Figure 1: Conditional generation with different factorization: *ind*(left) and *cond-ind*(right).

where $C(\alpha) = \int \exp(f_\alpha(z))\mathrm{d}z$ is the partition function and function $f_\alpha : \mathbb{R}^d \to \mathbb{R}$ is often parameterized as a multi-layer perceptron (MLP) with parameters α learned from observed data. However, it should be noted that Equation 1 defines a middle-ground model, where a Gaussian distribution is included as reference. In this work, we are interested in studying the behaviour of a *pure* EBM prior, where the Gaussian term is removed. We name this latter prior *EBM*, and the previous one *EBM-Gaussian*.

4.2 Factorization of $p_\beta(x, y|z)$

Having y as an attribute external to the latent space, there exists two approaches we can follow to model this interaction: the first assumes the independence between the variables Y and Z, while the latter one assumes the conditional independence of X and Y given Z. Intuitively, the first case forces z and y to be disentangled, while in the second case, z contains all the necessary information for the generation of the observation x. These two conflicting ways of modeling $p_\beta(y, x|z)$ can be reduced to a difference in the factorization of the associated probabilistic graphical model, which are represented in Figure 1. We write them as follows:

- *ind*: $p_\beta(x, y|z) = p_\beta(x|y, z) \times p_\beta(y)$
- *cond-ind*: $p_\beta(x, y|z) = p_\beta(x|z) \times p_\beta(y|z)$

They result in the following generation process: (1) The condition y is sampled from some fixed distribution $p(y)$; (2) In the case of *ind*, a latent continuous vector z is sampled from the distribution $p_\alpha(z)$; in the case of *cond-ind*, we sample z instead from un-parameterized posterior $p(z|y)$ with Langevin Monte Carlo (LMC), requiring the computation of $\nabla_z \log p_\theta(z|y)$, which can be solved with the help of $p_\alpha(z)$ and $p_\beta(y|z)$:

$$\nabla_z \log p_\theta(z|y) = \nabla_z \left[\log p_\alpha(z) + \log p_\beta(y|z)\right] \tag{2}$$

(3) Noting $u = \{z\}$ or $\{z, y\}$ depending on the factorization, the sequence x is sampled from the conditional distribution $p_\beta(x|u)$ which parametrization is usually referred to as *generator network*. With the observation x being a sequence of L words, the generator network takes the form of a conditional autoregressive model parameterized by a recurrent network, of parameters β, as follows:

$$p_\beta(x|u) = \prod_{l=1}^{L} \Phi_\beta(x^l | x^1, \cdots, x^{l-1}, u) \tag{3}$$

Thus, the generation process consists in successively sampling tokens x^l from the categorical distribution over the vocabulary, Φ_β. On the other hand, the distribution of attribute y, $p_\beta(y|z)$ is defined as categorical distribution parameterized by a MLP.

4.3 Learning algorithm

The latent-variable models described above are trained through Maximum-Likelihood estimation of the marginal density:

$$\min_\theta \mathbb{E}_{(x,y)\sim p_{\text{data}}} \left[-\log p_\theta(x, y)\right] \tag{4}$$

With the presence of latent variable Z, the log marginal likelihood is written as an intractable integral:

$$\log p_\theta(x, y) = \log \int_{\mathcal{Z}} p_\alpha(z) p_\beta(x, y|z)\mathrm{d}z \tag{5}$$

The integral presented in Equation 5 is often intractable when $\mathcal{Z}$ is high-dimensional and p_θ is parameterized by a neural network. The dominant surrogate approaches to optimizing this objective are Expectation Maximization (EM) and, more recently, Variational Inference (VI).

Expectation Maximization. The EM algorithm is an iterative procedure based on repeatedly optimizing the *expected complete data likelihood* given the current parameters. This quantity is computed in the E-step:

$$\begin{aligned}\mathbb{E}_{(x,y)\sim p_{\text{data}}} \Big[\mathbb{E}_{z\sim p_{\theta^t}(z|x,y)} \big[\log p_\alpha(z) \\ + \log p_\beta(x, y|z)\big]\Big]\end{aligned} \tag{6}$$

where θ^t is the estimate for θ at current step t. The inner expectation in Equation 6 can be further approximated through Monte Carlo (MC) estimation:

$$\frac{1}{P}\sum_{p=1}^{P}(\log p_\alpha(z_p) + \log p_\beta(x, y|z_p)) \quad (7)$$

where z_p denotes the samples drawn from the posterior distribution $p_{\theta^t}(z|x, y)$ estimated at the current step. In order to efficiently obtain these samples, we can use the LMC (Rossky et al., 1978; Parisi, 1981). [2] Then, in the M-step, we simply need to maximize the quantity in Equation 7 with respect to $\theta = \{\alpha, \beta\}$. With the Stochastic Gradient Descent (SGD), the M-step can be replaced by a single gradient update: it is indeed possible to prove that[3]

$$\nabla_\theta \log p_\theta(x, y) = \mathbb{E}_{z\sim p_\theta(z|x,y)} \nabla_\theta \log p_\theta(x, y, z) \quad (8)$$

which is in fact equal to the gradient of the inner expectation computed during the E-step.

Variational Inference. As it is impossible to compute the exact posterior $p_\theta(z|x, y)$, we can introduce an approximate posterior $q_\phi(z)$, which is called the *variational distribution* and is usually chosen to be a multivariate Gaussian with diagonal covariance: $q_\phi(z) = \mathcal{N}(z; \mu, \sigma^2\mathbf{I})$ with $\phi = \{\mu, \sigma^2\}$. As it is often done, we use Amortised Variational Inference (AVI) (Gershman and Goodman, 2014) scale up VI by learning a function g_γ that transforms each data point (x, y) into the parameters of the approximate posterior:

$$q_\phi(z) = \mathcal{N}(z; g_\gamma(x, y)) \quad (9)$$

where g_γ is often referred to as inference network, parameterized as a recurrent neural network in our case. We can then maximize a lower bound of $\log p_\theta(x, y)$, called the Evidence Lower Bound, or ELBO:

$$\mathbb{E}_{z\sim q_\phi(z)} [\log p_\beta(x, y|z)] - \mathbb{D}_{\mathrm{KL}}(q_\phi(z)||p_\alpha(z)) \quad (10)$$

The recent literature on VAEs usually employs the ELBO under the form shown in Equation 10, as the KL divergence can be computed analytically when both q_ϕ and p_α are Gaussians, which is not always the case in our framework. To facilitate the deduction of the surrogate loss functions for all our models, we rewrite the ELBO as follows:

$$\mathbb{E}_{z\sim q_\phi(z)} [\log p_\alpha(z) + \log p_\beta(x, y|z)] + \mathbb{H}(q_\phi(z)) \quad (11)$$

Then, an advantage of our particular choice of variational distribution is that it can provide a closed-form expression for several terms, among which the entropy:

$$\mathbb{H}(q_\phi(z)) = \frac{d}{2}\log(2\pi) + \frac{1}{2}\sum_{j=1}^{d}(1 + \log\sigma_j^2) \quad (12)$$

where d is the dimension of $\mathcal{Z}$.

4.4 Loss functions

To summarize, for each parameter update we need to compute, when using EM:

$$\nabla_{\alpha,\beta}\left[-\frac{1}{P}\sum_{p=1}^{P}(\log p_\alpha(z_p) + \log p_\beta(x, y|z_p))\right] \quad (13)$$

and the following gradient when using VI:

$$\begin{aligned}\nabla_{\alpha,\beta,\phi} [- \mathbb{E}_{z\sim q_\phi(z)} [\log p_\alpha(z) & \quad (14)\\ + \log p_\beta(x, y|z)] - \mathbb{H}(q_\phi(z))]\end{aligned}$$

In both cases, the computation of the gradient $\nabla_\alpha \log p_\alpha(z)$ can be problematic when we adopt an EBM as prior, since the partition function makes the computation of $\log p_\alpha(z)$ intractable. This leads us to expand the gradient as follows[4]:

$$\nabla_\alpha \log p_\alpha(z) = \nabla_\alpha f_\alpha(z) - \mathbb{E}_{z\sim p_\alpha(z)} [\nabla_\alpha f_\alpha(z)] \quad (15)$$

which can also be approximated through MC estimation:

$$\nabla_\alpha \log p_\alpha(z) \approx \nabla_\alpha \left[f_\alpha(z) - \frac{1}{Q}\sum_{q=1}^{Q} f_\alpha(z_q)\right] \quad (16)$$

where the z_i are the samples drawn from the distribution $p_\alpha(z)$ with the LMC algorithm. In addition, in the case of VI, since $q_\phi(z)$ and $q_\alpha(z)$ are chosen to be respectively $\mathcal{N}(\mu, \sigma^2)$ and $\mathcal{N}(0, \mathrm{I})$, we can simplify the related expectation by:

$$\mathbb{E}_{z\sim q_\phi(z)} [\log p_\alpha(z)] = \\ -\frac{d}{2}\log(2\pi) - \frac{1}{2}\sum_{j=1}^{d}(\mu_j^2 + \sigma_j^2) \quad (17)$$

We now have all the information needed for the computation of surrogates to the loss functions associated with the possible scenarios in our framework. Given the number of possibilities involved,

[2] See Appendix A for more details about the Langevin Monte Carlo algorithm and Appendix B for a description of its application to posterior sampling.

[3] See Appendix C for a detailed derivation.

[4] See chapter 18.1 of Goodfellow et al. (2016) for a detailed derivation.

Alg.	Fact.	Prior	Surrogate loss function
EM	*ind*	Gaussian	$\mathbb{E}_{(x,y)\sim p_{\text{data}}}\left[-\left(\frac{1}{P}\sum_{p=1}^{P}\log p_\beta(x\|y,z_p)+\log p_\beta(y)\right)\right]$
		EBM	$\mathbb{E}_{(x,y)\sim p_{\text{data}}}\left[-\left(\frac{1}{P}\sum_{p=1}^{P}(\log p_\beta(x\|y,z_p)+f_\alpha(z_p))-\frac{1}{Q}\sum_{q=1}^{Q}f_\alpha(z_q)+\log p_\beta(y)\right)\right]$
VI	*ind*	Gaussian	$\mathbb{E}_{(x,y)\sim p_{\text{data}}}\left[-\left[\mathbb{E}_{z\sim q_\phi(z)}\left[\log p_\beta(x\|y,z)+\log p_\beta(y)\right]+\frac{1}{2}\sum_{j=1}^{d}(\log\sigma_j^2-\sigma_j^2-\mu_j^2)\right]\right]$
		EBM	$\mathbb{E}_{(x,y)\sim p_{\text{data}}}\left[-\left[\mathbb{E}_{z\sim q_\phi(z)}\left[\log p_\beta(x\|y,z)+\log p_\beta(y)+f_\alpha(z)\right]-\frac{1}{Q}\sum_{q=1}^{Q}f_\alpha(z_q)+\frac{1}{2}\sum_{j=1}^{d}(\log\sigma_j^2)\right]\right]$
		EBM-G	$\mathbb{E}_{(x,y)\sim p_{\text{data}}}\left[-\left[\mathbb{E}_{z\sim q_\phi(z)}\left[\log p_\beta(x\|y,z)+\log p_\beta(y)+f_\alpha(z)\right]-\frac{1}{Q}\sum_{q=1}^{Q}f_\alpha(z_q)+\frac{1}{2}\sum_{j=1}^{d}(\log\sigma_j^2-\sigma_j^2-\mu_j^2)\right]\right]$
	cond-ind	Gaussian	$\mathbb{E}_{(x,y)\sim p_{\text{data}}}\left[-\left[\mathbb{E}_{z\sim q_\phi(z)}\left[\log p_\beta(x\|z)+\log p_\beta(y\|z)\right]+\frac{1}{2}\sum_{j=1}^{d}(\log\sigma_j^2-\sigma_j^2-\mu_j^2)\right]\right]$
		EBM	$\mathbb{E}_{(x,y)\sim p_{\text{data}}}\left[-\left[\mathbb{E}_{z\sim q_\phi(z)}\left[\log p_\beta(x\|z)+\log p_\beta(y\|z)+f_\alpha(z)\right]-\frac{1}{Q}\sum_{q=1}^{Q}f_\alpha(z_q)+\frac{1}{2}\sum_{j=1}^{d}(\log\sigma_j^2)\right]\right]$
		EBM-G	$\mathbb{E}_{(x,y)\sim p_{\text{data}}}\left[-\left[\mathbb{E}_{z\sim q_\phi(z)}\left[\log p_\beta(x\|z)+\log p_\beta(y\|z)+f_\alpha(z)\right]-\frac{1}{Q}\sum_{q=1}^{Q}f_\alpha(z_q)+\frac{1}{2}\sum_{j=1}^{d}(\log\sigma_j^2-\sigma_j^2-\mu_j^2)\right]\right]$

Table 1: Surrogate loss functions for the set of models experimented with in Section 5.

and especially because of the large computation time expected with the MC estimation in our EM algorithm, we explore a restricted set of combinations: we only compare both factorizations when learning with VI. We also only include the *EBM-Gaussian* prior with VI, as it is supposed to mitigate the fact that and EBM prior does not fit the assumption made by VI. This set, and the list of associated surrogate loss functions, is detailed in Table 1.

5 Application to conditional text generation

5.1 Experimental setup

Datasets. To evaluate our models on conditional text generation, we use two datasets: *Yelp* and *News Aggregator* (Dua and Graff, 2017). *Yelp* consists of restaurant reviews; we use the version preprocessed by Shen et al. (2017) which includes only two polarity sentiment labels (*Positive* and *Negative*) and sentences that are no longer than 15 words. *News Aggregator* is a collection of news articles from four categories: *Business*, *Sci-tech* (science and technology), *Entertainment* and *Health*. We use only the titles of the articles for text generation: in this setting, the dataset is usually referred to as *News Titles*[5].

Evaluation metrics. In this paper, we consider the following aspects: the realism of the generated sentences, their diversity, and the ability of the model to control sentence attribute. For the realism of sentences, and their diversity, we use respectively Forward BLEU and Backward BLEU, which were first proposed by Shi et al. (2018) for the evaluation of unconditional text generation. Forward BLEU computes the BLEU score (Papineni et al., 2002) of each generated text by using the whole test set as reference and takes the average, while Backward BLEU takes all the generated sentences as reference and computes the BLEU score of each test sentence. In order to evaluate the ability of the model to control sentence attribute, we use a FastText (Joulin et al., 2017) classifier[6] which will measure how consistent are the attributes of our generated sentences. We pre-train the FastText classifier on real-world data and then use it as an oracle classifier. We reserve a subset of the training data exclusively in order to pre-train the classifier, and not to be used to train the generative model[7]. Our FastText classifier achieves accuracies of respectively 96.7% and 91.3% on *Yelp* and *News Titles*. To easily summarize results, we compute the geometric mean of these three metrics. When evaluating, for each dataset, model, and possible attribute, we generate the same number of sentences as there is in the associated test set.

Details on models and optimization[8]**.** We model the EBM scoring function f_α using a MLP with GELU activations and the generator p_β with a GRU (Cho et al., 2014) with one hidden layer. In the case of the *ind* factorization, we represent the attribute y with a one-hot encoding and use it both to initialize the hidden state of the GRU, and concatenated to the word embedding inputs. Concerning the optimization of models with EBM priors, we add a hyperparameter λ for weighting the EBM-term in the loss function, in order to be able to stabilize the training. For each different experiment, the hyper-parameters were searched with

[5] See Appendix D for more details about these two datasets.

[6] See Appendix E for more details on the oracle classifier.

[7] Details on datasets splits can be found in Appendix D.

[8] Further details about the optimization, regularization, and hyper-parameter search can be found in Appendix F.

Dataset	Algorithm	Factorization	Prior	Acc ↑	F-BLEU ↑	B-BLEU ↑	G-mean ↑
Yelp	EM	*ind*	Gaussian	0.9852	0.5816	0.2369	0.5139
			EBM	0.9574	0.8254	0.3396	0.6450
	VI	*ind*	Gaussian	0.9786	0.8074	0.4476	0.7072
			EBM	0.9646	0.8010	0.3763	0.6625
		cond-ind	Gaussian	0.9046	0.8327	0.4286	0.6860
			EBM	0.8771	0.6451	0.3307	0.5719
			EBM-Gaussian	0.9066	0.8428	0.4157	0.6823
News Titles	VI	*ind*	Gaussian	0.9547	0.4892	0.2210	0.4690
			EBM	0.9119	0.4309	0.1778	0.4119
		cond-ind	Gaussian	0.7683	0.4576	0.2197	0.4259
			EBM	0.6971	0.3382	0.1322	0.3147
			EBM-Gaussian	0.8376	0.5001	0.2056	0.4416

Table 2: Conditional text generation results on *Yelp* and *News Titles*. Experiments with EM on *News Titles* were not included because of the large runtime required by posterior inference.

a random search strategy (Bergstra and Bengio, 2012), with 16 runs.

5.2 Results

We base model selection on the G-mean of our three metrics computed on the validation set, and present the performance of the selected models[9] on test sets in Table 2. First, we observe that when training the model with the EM algorithm on *Yelp*, an EBM prior results in a substantial improvement of the conditional generative performance with respect to all the metrics. This seems to confirm the hypothesis that an EBM can learn a more flexible prior, which increases the representational expressivity[10]. However, with VI, the models based on an EBM prior perform worse. We conjecture that the reason is two-fold: firstly, training an EBM on high-dimensional data is difficult, and all the more when the EBM is moved into latent-space; we will develop this point in Section 6.1. Secondly, with VI, we make a Gaussian assumption on the posterior distribution; minimizing the KL divergence between the posterior and the prior is hence restrictive when learning an EBM-based prior. This explains the better performance of the EBM-Gaussian prior, which is almost identical to the Gaussian prior. Finally, a comparison of the results for both factorizations shows that the models based on *ind* generally perform better in classification accuracy than those based on *cond-ind*. A possible explanation is that the posterior LMC sampling $z \sim p_\theta(z|y)$ necessary for conditional generation adds a supplementary difficulty to the process, since the LMC sampling hardly converges in high dimension. Overall, the model variant [*VI* • *ind* • *Gaussian*] (i.e, S-VAE) is the best performing in our framework, on both datasets.

6 Discussion

6.1 Training of EBM-based priors

Despite their flexibility for generative modeling, EBMs are notorious for their unstable training, especially when it comes to high dimensional spaces. When the modeling takes place directly in the observed data space, previous works (Xie et al., 2016; Du and Mordatch, 2019; Grathwohl et al., 2020) using LMC on EBMs observed that short-run LMC chains with a Contrastive Divergence (CD) or Persistent CD initialization can eventually generate realistic samples, even though the model has not converged[11]. However, moving an EBM into the latent space introduces additional components to the loss to be optimized, and this difficulty to converge can no longer be ignored: other parts of the

[9]See Appendix H for samples of sentences generated by the different models.

[10]However, we could not confirm those results in a reasonable time on *News Titles*, given the large runtime required by the posterior inference.

[11]The recent work of Nijkamp et al. (2020) shows that despite this, the energy of a trained EBM which has not converged does not necessarily approximate the real density well.

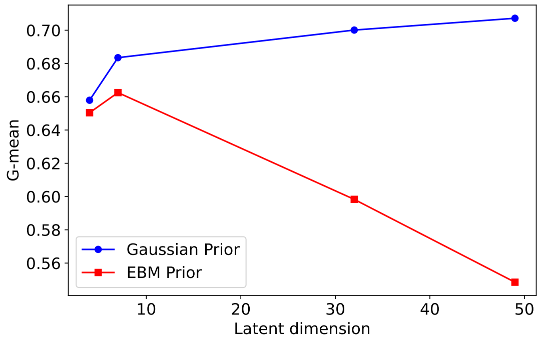

Figure 2: Influence of dimension of the latent space on the G-mean for models with gaussian and EBM-based priors trained with VI.

loss can easily be affected as they are optimized jointly. It forces us to try to circumvent the issue through strategies such as setting a small weight for the EBM-related term of the loss, or diminishing the dimension of the latent space. While these solutions allowed the training to stabilize, they in turn slow the learning of the prior and limit its expressive ability. This is very significant with VI[12], as we can see on Figure 2: the performance of the gaussian prior increases with the latent dimension, while the performance of the EBM prior plummets.

6.2 Comparison with SVEBM

The Symbol-Vector Coupling Energy-Based Model (SVEBM) of Pang and Wu (2021) uses both an EBM-based prior and employs VI to approximate the posterior distribution for the application of attribute-controlled text generation. Within our framework, the variant [*VI • Cond-ind • EBM-Gaussian*] is the most similar to the SVEBM: we will from now on refer to it as **VCEG**. However, differently from our separate parameterizations of prior $p_\alpha(z)$ and classifier $p_\beta(y|z)$, the SVEBM formulates the joint distribution $p_\alpha(y, z)$ as an EBM $p_\alpha(y, z) = \frac{1}{C(\alpha)} \exp(\langle y, f_\alpha(z)\rangle) \times \mathcal{N}(z; 0, I)$ .As such, it can be seen as using a Joint Energy-based Model (JEM) (Grathwohl et al., 2020) in the latent space. In addition, Pang and Wu (2021) proposes to improve learning with a regularization mechanism based on the information bottleneck (SVEBM-IB). However, on attribute-controlled text generation, Pang and Wu (2021) only report the accuracy of an oracle classifier on generated sentences for the SVEBM-IB, leaving out the base model. To obtain a more complete picture, we compare in Table 3 the performances of the related models with respect of our three metrics. VCEG obtains the best performance among all the models. However, our implementation of the SVEBM performs slightly worse than the original implementation. Still, comparing the first two rows clearly shows us that using joint energy modeling in the latent space harms the controllability of the model, rendering necessary the information bottleneck trick, which, in turns, reduces its expressivity.

Model	Acc	F-BLEU	B-BLEU	G-mean
VCEG†	0.9066	0.8428	0.4157	0.6823
SVEBM†	0.8206	0.7624	0.3858	0.6226
SVEBM‡	0.7590	0.8296	0.4406	0.6522
SVEBM-IB‡	0.8580	0.8912	0.3782	0.6613

Table 3: Performance of the SVEBM-related models on the *Yelp* dataset. † refers to our own implementation. ‡ refers to the implementation of Pang and Wu (2021) [13].

7 Conclusion

In this work, we have sought to clarify how several key factors in the design of latent-variable generative models (complexity of the prior, interaction between the attribute and the latent representation, learning method) affect their performance on conditional text generation tasks. We experiment in particular with EBM-based priors, and show that while these priors indeed have greater representational power than the usual Gaussian priors, they are currently hard to exploit on account of their problematic training. Our experiments also show that coupling attribute and latent variable, as done in the SVEBM (Pang and Wu, 2021) is not an optimal solution. Finally, in our unified framework, we observe that the best performing model remains the earliest, corresponding to the design of the S-VAE (Kingma et al., 2014).

References

James Bergstra and Yoshua Bengio. 2012. Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, 13(10):281–305.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016.

[12]These solutions were not required with EM: see Appendix F for the details of the latent dimensions selected by the hyper-parameter search in each setting.

[13]We used the official code released by the authors: `https://github.com/bpucla/ibebm` (git commit: 315b645). To get the results of SVEBM, we removed the term of the loss corresponding to the information bottleneck.

Generating Sentences from a Continuous Space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany.

Sophie Burkhardt and Stefan Kramer. 2019. Decoupling sparsity and smoothness in the dirichlet variational autoencoder topic model. *Journal of Machine Learning Research*, 20(131):1–27.

Khyathi Chandu, Shrimai Prabhumoye, Ruslan Salakhutdinov, and Alan W Black. 2019. "my way of telling a story": Persona based grounded story generation. In *Proceedings of the Second Workshop on Storytelling*, pages 11–21, Florence, Italy. Association for Computational Linguistics.

Xi Chen, Diederik P. Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2017. Variational Lossy Autoencoder. In *International Conference on Learning Representations*, Toulon, France. OpenReview.net.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc Le, and Christopher D. Manning. 2020. Pre-training transformers as energy-based cloze models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 285–294, Online. Association for Computational Linguistics.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*.

Yuntian Deng, Anton Bakhtin, Myle Ott, Arthur Szlam, and Marc'Aurelio Ranzato. 2020. Residual energy-based models for text generation. In *International Conference on Learning Representations*.

Xiaoan Ding and Kevin Gimpel. 2021. FlowPrior: Learning expressive priors for latent variable sentence models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3242–3258, Online. Association for Computational Linguistics.

Yilun Du and Igor Mordatch. 2019. Implicit Generation and Modeling with Energy Based Models. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository.

Yu Duan, Canwen Xu, Jiaxin Pei, Jialong Han, and Chenliang Li. 2020. Pre-train and Plug-in: Flexible Conditional Text Generation with Variational Auto-Encoders. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 253–262, Online. Association for Computational Linguistics.

Simon Duane, A. D. Kennedy, Brian J. Pendleton, and Duncan Roweth. 1987. Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222.

Scott E. Fahlman, Geoffrey E. Hinton, and Terrence J. Sejnowski. 1983. Massively Parallel Architectures for AI: Netl, Thistle, and Boltzmann Machines. In *Proceedings of the Third AAAI Conference on Artificial Intelligence*, AAAI'83, pages 109–113, Washington, D.C. AAAI Press.

Le Fang, Chunyuan Li, Jianfeng Gao, Wen Dong, and Changyou Chen. 2019. Implicit deep latent variable models for text generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3946–3956, Hong Kong, China. Association for Computational Linguistics.

Stuart Geman and Donald Geman. 1984. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741.

Samuel Gershman and Noah D. Goodman. 2014. Amortized Inference in Probabilistic Reasoning. In *Proceedings of the 36th Annual Meeting of the Cognitive Science Society*, Quebec City, Canada. cognitivesciencesociety.org.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy. PMLR.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.

Will Grathwohl, Kuan-Chieh Wang, Joern-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. 2020. Your Classifier is Secretly an Energy Based Model and You Should Treat it Like One. In *International Conference on Learning Representations*.

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings* *of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 297–304, Chia Laguna Resort, Sardinia, Italy. PMLR.

W Keith Hastings. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109.

Tianxing He, Bryan McCann, Caiming Xiong, and Ehsan Hosseini-Asl. 2021. Joint energy-based model training for better calibrated natural language understanding models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1754–1761, Online. Association for Computational Linguistics.

Geoffrey E. Hinton. 2002. Training Products of Experts by Minimizing Contrastive Divergence. *Neural Computation*, 14(8):1771–1800.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward Controlled Generation of Text. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1587–1596, Sydney, Australia. PMLR.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of Tricks for Efficient Text Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

Nitish Shirish Keskar, Bryan McCann, Lav Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL - A Conditional Transformer Language Model for Controllable Generation. *arXiv preprint arXiv:1909.05858*.

Yoon Kim, Sam Wiseman, Andrew Miller, David Sontag, and Alexander Rush. 2018. Semi-Amortized Variational Autoencoders. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2678–2687, Stockholmsmässan, Stockholm Sweden. PMLR.

Diederik P Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations*.

Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. 2014. Semi-supervised Learning with Deep Generative Models. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Yuan Li, Chunyuan Li, Yizhe Zhang, Xiujun Li, Guoqing Zheng, Lawrence Carin, and Jianfeng Gao. 2020. Complementary Auxiliary Classifiers for Label-Conditional Text Generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8303–8310.

Yang Lu, Song-Chun Zhu, and Ying Wu. 2016. Learning FRAME Models Using CNN Filters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. 1953. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6):1087–1092.

Radford M Neal and others. 2011. MCMC using Hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2.

Erik Nijkamp, Mitch Hill, Tian Han, Song-Chun Zhu, and Ying Nian Wu. 2020. On the Anatomy of MCMC-Based Maximum Likelihood Learning of Energy-Based Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):5272–5280. AAAI.

Bo Pang, Tian Han, Erik Nijkamp, Song-Chun Zhu, and Ying Nian Wu. 2020. Learning Latent Space Energy-Based Prior Model. In *Advances in Neural Information Processing Systems*, volume 33, pages 21994–22008. Curran Associates, Inc.

Bo Pang and Ying Nian Wu. 2021. Latent space energy-based model of symbol-vector coupling for text generation and classification. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8359–8370. PMLR. ICML.

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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

G. Parisi. 1981. Correlation functions and computer simulations. *Nuclear Physics B*, 180(3):378–384.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1310–1318, Atlanta, Georgia, USA. PMLR.

Tom Pelsmaeker and Wilker Aziz. 2020. Effective Estimation of Deep Generative Language Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7220–7236, Online. Association for Computational Linguistics.

Shrimai Prabhumoye, Alan W Black, and Ruslan Salakhutdinov. 2020. Exploring Controllable Text Generation Techniques. In *Proceedings of the 28th*

International Conference on Computational Linguistics, pages 1–14, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1278–1286, Bejing, China. PMLR.

Ronald Rosenfeld, Stanley F. Chen, and Xiaojin Zhu. 2001. Whole-sentence exponential language models: a vehicle for linguistic-statistical integration. *Computer Speech & Language*, 15(1):55–73.

P. J. Rossky, J. D. Doll, and H. L. Friedman. 1978. Brownian dynamics as smart Monte Carlo simulation. *The Journal of Chemical Physics*, 69(10):4628–4633.

Ruslan Salakhutdinov and Geoffrey Hinton. 2009. Deep Boltzmann Machines. In *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 448–455, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA. PMLR.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems*, volume 30, pages 6830–6841. Curran Associates, Inc. NIPS.

Wenxian Shi, Hao Zhou, Ning Miao, and Lei Li. 2020. Dispersed exponential family mixture vaes for interpretable text generation. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 8840–8851. PMLR.

Zhan Shi, Xinchi Chen, Xipeng Qiu, and Xuanjing Huang. 2018. Toward Diverse Text Generation with Inverse Reinforcement Learning. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pages 4361–4367. International Joint Conferences on Artificial Intelligence Organization.

P. Smolensky. 1986. Information Processing in Dynamical Systems: Foundations of Harmony Theory. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*, pages 194–281. MIT Press, Cambridge, MA, USA.

Tijmen Tieleman. 2008. Training Restricted Boltzmann Machines Using Approximations to the Likelihood Gradient. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 1064–1071, New York, NY, USA. Association for Computing Machinery.

Jakub Tomczak and Max Welling. 2018. VAE with a VampPrior. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 1214–1223. PMLR.

Lifu Tu, Richard Yuanzhe Pang, Sam Wiseman, and Kevin Gimpel. 2020. ENGINE: Energy-based inference networks for non-autoregressive machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2819–2826, Online. Association for Computational Linguistics.

Bin Wang and Zhijian Ou. 2017. Language modeling with neural trans-dimensional random fields. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 294–300.

Bin Wang, Zhijian Ou, and Zhiqiang Tan. 2015. Trans-dimensional Random Fields for Language Modeling. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 785–794, Beijing, China. Association for Computational Linguistics.

Wenlin Wang, Zhe Gan, Hongteng Xu, Ruiyi Zhang, Guoyin Wang, Dinghan Shen, Changyou Chen, and Lawrence Carin. 2019. Topic-guided variational auto-encoder for text generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 166–177, Minneapolis, Minnesota. Association for Computational Linguistics.

Jianwen Xie, Yang Lu, Song-Chun Zhu, and Yingnian Wu. 2016. A Theory of Generative ConvNet. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2635–2644, New York, New York, USA. PMLR.

Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. 2017. Improved Variational Autoencoders for Text Modeling using Dilated Convolutions. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3881–3890, International Convention Centre, Sydney, Australia. PMLR.

Junbo Zhao, Yoon Kim, Kelly Zhang, Alexander Rush, and Yann LeCun. 2018. Adversarially Regularized Autoencoders. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5902–5911, Stockholmsmässan, Stockholm Sweden. PMLR. ICML.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. Learning Discourse-level Diversity for Neural Dialog Models using Conditional Variational Autoencoders.

In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–664, Vancouver, Canada. Association for Computational Linguistics.

Song Chun Zhu, Yingnian Wu, and David Mumford. 1998. Filters, Random Fields and Maximum Entropy (FRAME): Towards a Unified Theory for Texture Modeling. *International Journal of Computer Vision*, 27(2):107–126.

A Langevin Monte Carlo

Let π be a target density distribution, expressed as:

$$\pi(x) = \mathrm{e}^{-U(x)} / \int_{\mathbb{R}^d} \mathrm{e}^{-U(y)} \mathrm{d}y \tag{18}$$

where $U : \mathbb{R}^d \to \mathbb{R}$; sampling from π can be achieved through MCMC methods, such as Hastings-Metropolis algorithm (Metropolis et al., 1953; Hastings, 1970), Gibbs sampling (Geman and Geman, 1984) or Hamiltonian Monte Carlo (Duane et al., 1987; Neal and others, 2011). LMC (also called Unadjusted Langevin Algorithm) proposes to construct the Markov chain $(X^k)_{k\geq 0}$ given for all $k \in \mathbb{N}$ by:

$$X^{k+1} = X^k - \lambda \nabla U(X^k) + \sqrt{2\lambda} G^{k+1} \tag{19}$$

where $\lambda > 0$ is the constant stepsize and $(G^k)_{k\geq 1}$ is a sequence of i.i.d. standard d-dimensional Gaussian vectors. In fact, LMC is a special case of Metropolis-Hastings algorithm by taking the proposal distribution $\mathcal{N}(X_k - \lambda \nabla U(X_k), \sqrt{2\lambda}\mathbf{I}_d)$. To avoid long Markov chain mixing time, and reduce significantly the numbers of steps necessary to converge, Contrastive Divergence (CD) (Hinton, 2002) takes the data samples as initial states while Persistent Contrastive Divergence (PCD) (Tieleman, 2008) takes instead the negative samples generated by the model distribution in the previous learning step; in this work, we use the latter.

B LMC for posterior sampling

In order to sample from $p_\theta(z|x,y)$ with LMC, we can rewrite $p_\theta(z|x,y) = \exp(\log p_\theta(z|x,y))$ in the form of EBM, considering $\log p_\theta(z|x,y)$ as the energy function. The calculation of $\nabla_z \log p_\theta(z|x,y)$ is thus involved when applying LMC:

$$\begin{aligned}
\nabla_z \log p_\theta(z|x,y) &= \nabla_z \log \frac{p_\theta(x,y,z)}{p_\theta(x,y)} \\
&= \nabla_z \log p_\theta(x,y,z) \\
&= \nabla_z \log p_\beta(x,y|z) \times p_\alpha(z) \\
&= \nabla_z \log p_\beta(x,y|z) + \nabla_z \log p_\alpha(z) \\
&= \nabla_z \log p_\beta(x,y|z) + \nabla_z \log f_\alpha(z) - \nabla_z \log C(\alpha) \\
&= \nabla_z \log p_\beta(x,y|z) + \nabla_z \log f_\alpha(z) \qquad (20)
\end{aligned}$$

where $p_\beta(x,y|z)$ and $f_\alpha(z)$ can be computed by conducting the forward propagation of the neural network.

C Deduction of Equation 8

Taking the gradient of the single log-likelihood, we have

$$\begin{aligned}
\nabla_\theta \log p_\theta(x,y) &= \log p_\theta(x,y) \int q_\lambda(z) dz \\
&= \int q_\lambda(z) \nabla_\theta \log p_\theta(x,y) dz \\
&= \mathbb{E}_{q_\lambda(z)} \nabla_\theta \log p_\theta(x,y) \\
&= \mathbb{E}_{q_\lambda(z)} \nabla_\theta \log \frac{p_\theta(x,y,z)}{p_\theta(z|x,y)} \\
&= \mathbb{E}_{q_\lambda(z)} [\nabla_\theta \log p_\theta(x,y,z) - \nabla_\theta \log p_\theta(z|x,y)]
\end{aligned}$$

Since $\mathbb{E}_{p_\theta(z|x,y)} \nabla_\theta \log p_\theta(z|x,y) = 0$, taking $p_\theta(z|x,y)$ as $q_\lambda(z)$, we have:

$$\nabla_\theta \log p_\theta(x,y) = \mathbb{E}_{p_\theta(z|x,y)} \nabla_\theta \log p_\theta(x,y,z)$$

D Additional details about datasets

We have carried out experiments on three text datasets: *Yelp* [14] and *News Titles* [15], and lastly, *Name* [16]. *Yelp* dataset is a subset of Yelp's businesses, reviews, and user data, originally provided by Yelp Dataset Challenge [17]. Multiple preprocessed versions exist for different purpose. We use the one processed by Shen et al. (2017), which contains two sentiment labels (negative and positive) and reviews no longer than 15 words. *News*

[14] Link to downloadable dataset: https://github.com/shentianxiao/language-style-transfer/tree/master/data/yelp

[15] Link to downloadable dataset: https://archive.ics.uci.edu/ml/datasets/News+Aggregator

[16] Link to downloadable dataset: https://github.com/spro/practical-pytorch/tree/master/data/names.

[17] https://www.yelp.com/dataset

Dataset	Attributes	Oracle classifier		Generative models		
		n_{train}	$n_{\text{validation}}$	n_{train}	$n_{\text{validation}}$	n_{test}
Name (Toy)	french	55	26	117	38	41
	dutch	65	26	124	42	40
Yelp	negative	31701	3519	141567	25278	50278
	positive	48237	5364	213713	38205	76392
News Title	business	20838	2260	74278	9257	9334
	science and technology	19545	2193	69409	8759	8597
	entertainment	27582	3036	97752	12165	12293
	health	8163	970	29241	3654	3611

Table 4: Statistics of datasets used in experiments

Titles (Dua and Graff, 2017) it should be noted that the version of used in our experiments is different than the one in Duan et al. (2020). We don't filter out titles longer than 15 words and we keep also *Science and Technology* category for the experiments, which retains the complexity of the origin dataset. Lastly, *Name* dataset is a collection of names from 18 languages of origin. We select French names and Dutch names among them to build a dataset with only two classes. We use it as a "toy dataset" for supplementary experiments and vizualisations of the learned density in latent space, shown in Appendix G. We present the data splitting details of all the datasets in Table 4.

E Oracle classifier

We utilize the FastText (Joulin et al., 2017) classifier to evaluate the generated sentences of all the models in our experiments. FastText is a linear classfier with word embeddings, updated at training time. A bag of n-grams is used as additional feature during the training. The choice of FastText is natural: it's efficient for both training and prediction with a reasonably accuracy. It can be trained on more than one billion words in less than ten minutes using a standard multicore CPU, and classify half a million sentences among 312K classes in less than a minute (Joulin et al., 2017). Besides, its simple model architecture makes it sharing less similarity with the generative model it is used to evaluate. The training hyper-parameters were not heavily tuned; we present them in Table 5.

Hyper-parameter	Name (Toy)	YELP	News Title
Training epochs	43	26	50
Learning rate	1.0	0.16	0.5
Word n-grams	5	3	3

Table 5: Hyper-parameters details for oracle classifier

F Hyper-parameters of generative models

In all our models, input embeddings are initialized with the Glorot normal initializer (Glorot and Bengio, 2010). For Yelp and News Titles, For all the model variants in our framework, we use one-layer bidirectional GRU of hidden dimension of 512 for both decoder and encoder when VI is employed. We parametrize the classifiers $p_\beta(y|z)$ and EBMs $p_\alpha(z)$ with MLPs of two hidden-layers of dimension 256 except for the EBM on *News Titles* where the number of hidden layer is set to one. The word dimension is set to 256 for all the experiments. As for the training, we train the models with a batch size of 128 and with an Adam optimizer of $\beta_1 = 0.9$, $\beta_2 = 0.9$ and $\epsilon = 1 \times 10^{-8}$. Concerning regularization, we adopt weight annealing for the regularization of the KL divergence $\frac{1}{2}\sum_{j=1}^{d}(\log \sigma_j^2 - \sigma_j^2 - \mu_j^2)$ and entropy $\frac{1}{2}\sum_{j=1}^{d}(\log \sigma_j^2)$. We also employ weight decay (L_2 penalty) to help regularization and gradient clipping (Pascanu et al., 2013) to deal with the exploding gradient problem. The coefficient of L_2 penalty is set to 0.1 while the maximum norm for the gradient clipping is set to 1. Other hyper-parameters are searched by random search strategy (Bergstra and Bengio, 2012) with the following distributions:

- We chose a dimension of latent space from $[\![1, 128]\!]$ uniformly.
- We chose a learning rate log-uniformly from 10^{-5} to 10^{-2}.

- We chose a number of LMC update step from $[\![31, 150]\!]$ uniformly.
- We chose a LMC step-size log-uniformly between 10^{-3} and 10.
- We chose a weight coefficient for EBM loss log-uniformly from 10^{-8} to 10^{-5}. The reason for this choice of search space is the fact that a large EBM weight loss will let the model diverge quickly, with extreme detriment to model performance, which can be observed in Figure 3.
- We chose a word dropout rate uniformly from $[0, 0.5]$.
- We chose a number of annealing step from $[\![1, 20000]\!]$ uniformly.

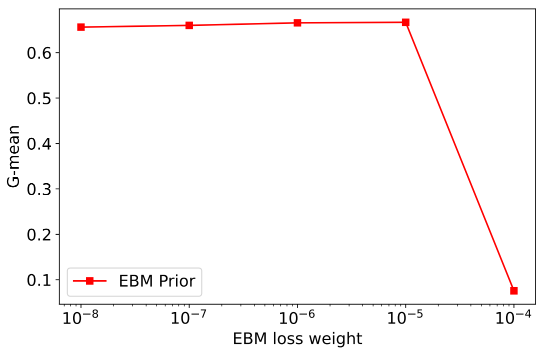

Figure 3: Influence of the dimension of the latent space on the G-mean for models with an EBM-based prior, trained with VI.

We conducted 16 trails of experiments for the search of hyper-parameters, for each model. The number of training steps were chosen with the early stopping strategy. For all the datasets used by our model in the experiments, those hyper-parameters of the best performance on the validation set can be found in Table 6.

G Visualization of learned latent space by EBM

In order to study further the behaviour of EBM in the latent space, we experiment on a simple (toy) dataset, *Name*, for which a 2-dimensional latent space is enough. Visualisation of the latent densities learned by different models with an EBM prior, shown in Figure 4, allows us to confirm that the distribution learned are in this case very distinct from the isotropic Gaussian distribution $\mathcal{N}(0, I_2)$. Additional quantitative results on this dataset are detailed in Table 7.

H Generated sentences samples

We present the sentences samples generated by different models in Table 8 and Table 9.

Dataset	Algo	Facto	Prior	dimension	learning rate	LMC n_{step}	LMC step-size	EBM weight	word dropout	$n_{\text{annealing}}$	n_{training}
Yelp	EM	*ind*	Gaussian	65	0.00105057	120	0.00239995	—	0.23167361	—	8000
			EBM	123	0.00411451	92	0.01358318	2.1e-06	0.06516866	—	2000
	VI	*ind*	Gaussian	49	0.000579	—	—	—	0.068292	17399	16000
			EBM	7	0.0008852	58	0.09894774	4.0e-08	0.28912746	13460	8000
		cond-ind	Gaussian	15	0.001114	149	0.003522	—	0.147971	17258	16000
			EBM	7	0.0008852	58	0.09894774	4.0e-08	0.28912746	13460	2000
			EBM-Gaussian	72	0.00044611	141	0.01327672	6.0e-08	0.25625266	19128	20000
News Titles	VI	*ind*	Gaussian	49	0.000579	—	—	—	0.068292	17399	20000
			EBM	7	0.0008852	58	0.09894774	4.0e-08	0.28912746	13460	18000
		cond-ind	Gaussian	15	0.001114	149	0.003522	—	0.147971	17258	20000
			EBM	7	0.0008852	58	0.09894774	4.0e-08	0.28912746	13460	2000
			EBM-Gaussian	7	0.0008852	58	0.09894774	4.0e-08	0.28912746	13460	18000

Table 6: Hyper-parameters for the generative models

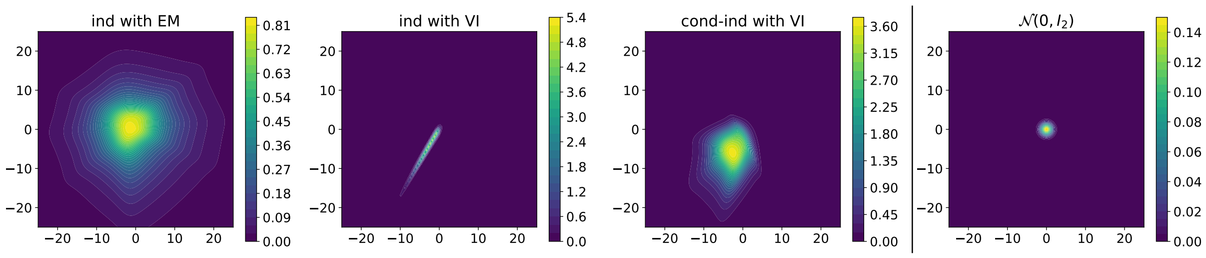

Figure 4: **Left:** Energy functions $\exp(f_\alpha(z))$ (proportional to density) of the latent space $\mathcal{Z}$ learned by different EBM variants in our framework. **Right:** Probability density of $\mathcal{N}(0, I_2)$.

Dataset	Learning algorithm	Factorization	Prior	Acc	F-BLEU	B-BLEU	G-mean
Name	EM	*ind*	Gaussian	0.9984	0.5083	0.1344	0.4086
			EBM	0.8275	0.3866	0.5418	0.5576
	VI	*ind*	Gaussian	0.8594	0.3778	0.5410	0.5600
			EBM	0.8314	0.3965	0.4055	0.5113
			EBM-Gaussian	0.9611	0.6239	0.2206	0.5095
		cond-ind	EBM	0.8126	0.3657	0.5572	0.5491

Table 7: Conditional text generation results on Name.

Algo.	Facto.	Prior	Attribute	Sentence samples
EM	*ind*	Gaussian	Positive	thanks chapel for your expertise ! i love this place ! great meal . amazing !
			Negative	i was so disappointed . worst apartment cleaners i 've ever been to . unfortunately i 'm not going back . do n't waste your time .
		EBM	Positive	i love this place ! it 's just very clean and the staff is very nice . the service is always great and the food is always fresh . so , i will not recommend this place .
			Negative	so , i will not recommend this place . i 'm not sure that i will not be back . i 'm not sure that they have been to least . the food was mediocre and the service was terrible .
VI	*ind*	Gaussian	Positive	what a great place . and if you want to be a regular , this is a great place . they take care of their customers to make their own and feel very comfortable . staff is friendly and the staff is always friendly and helpful .
			Negative	i ordered _num_ , and _num_ minutes for the first time . they have_num_ people in my office and i never return . my experience was taken off to our order . we will not be coming back for a few years .
		EBM	Positive	i was so happy with . i recommend the food and the food and they have always been great . it is a very good experience with a smile . the eggs benedict is also good and too .
			Negative	the chicken was not a good thing to have ever had . it was cooked and it was not cooked and tough . customer service was horrible . i gave the _num_ % of the reviews and they were .
	cond-ind	Gaussian	Positive	its always a nice place to get a date . the owner is a great guy and has a great attitude . this is the best , fast , and delicious . the sauce was perfect , and the sauce was very good .
			Negative	i asked for a new car and she said it was n't too busy . i am not sure to this place . avoid this place at all ! the only thing on the menu is good , but the food is very overpriced .
		EBM	positive	it was all of it was perfect . highly recommend . happy hour the service ! recommend this place , hands down .
			Negative	we ordered a salad and it was pretty good . but not really much good ! also , too , and no , and no sense of a smile . the worst part is the worst experience .
		EBM-Gaussian	positive	this is the worst i 've ever been to in my life . overall , a very good experience . great time to start with the service . they have great food and the service is friendly .
			Negative	at the end of the place i could have been to _num_ minutes . worst pizza hut i have ever had in a while . i would not recommend it . i could n't even eat it to eat .

Table 8: Sentence samples generated by conditioning on sentiment attribute. The models are trained on Yelp. The sentences are random selected with the help of RANDOM.SHUFFLE().

Algo.	Facto.	Prior	Attribute	Sentence samples
VI	*ind*	Gaussian	Business	fed's fisher to end up, but not to be strengthened barclays shunned by fitch european stocks rise ahead of yellen testimony
			Sci-tech	apple to unveil a new smart home platform for the next week windows phone 8.1 update with android 4.4.2 update and cortana support first look at the new android wear
			Entertain	lady gaga's tony bennett album release date, plus more details emerge a 'mrs. doubtfire' sequel in 'star wars: episode vii' is not a sequel jada pinkett smith: 'covert pedophiles' over willow smith
			Health	red robin thicke's new album in the works with new video duval county, other health care tips for global warming study: diabetic heart attacks, strokes falling
		EBM	Business	update: mothercare rejects takeover bid for astrazeneca's takeover offer warren buffett's berkshire pay gap in talks with astrazeneca disney buys klout for $280 million
			Sci-tech	ohio's state's ceo says google glass to be affected by... hon hai, pegatron on apple, ibm, and other tech giants how to watch the empire state building, and the world wide web?
			Entertain	rob kardashian and justin bieber and t.i. brawl in vegas brawl over t.i. brawl over t.i. brawl kim kardashian and kanye west one of thrones: george rr martin's new chapter
			Health	ohio state's first class seat to save lives officials: 1.8m pounds of ground beef products, including west africa exact sciences' deep-c data on cobimetinib
	cond-ind	Gaussian	Business	malaysia airlines flight 370 pilot flying down justin bieber caught in deposition video us supreme court rules against aereo in court
			Sci-tech	microsoft surface mini 2: surface pro 3 update: american apparel ceo dov charney's termination letter to american apparel apple iphone 6 rumors: 5.5-inch iphone 6 screens to enter production
			Entertain	rolf harris' disguised as' as he's' sickened 'by 18-year-old khloe kardashian and french montana embrace family feud prince harry and cressida bonas are dating, but dating?
			Health	why we should not trust care about tobacco sa news briefs nintendo apologizes for 'misleading' loss
		EBM	Business	us sanctions alibaba's ipo: amazon to buy the ipo the irs: astrazeneca's' to pay 'astrazeneca' in china's...
			Sci-tech	at & t's ceo's new york, the new york, and the new... best (ipad) samsung galaxy s5 price for india, price and gear 2...
			Entertain	'how i met your mother finale is the finale is the first time you need you need?? netflix ceo to $100 million in the us, but it's new york, but it's... fcc prices continue to be on again
			Health	los angeles attorney foods, says it's $1 million in new york... us county county county's death toll to continue to... nintendo posts $10.2bn million loss of $3.8 billion
		EBM-Gaussian	Business	us economy to grow up by 2.9% in first quarter, but still critical to... at & t agrees to buy directv for $48.5bn deal update 1-valeant shares soar after sycamore partners with verizon
			Sci-tech	us supreme court rules on aereo, 'right to be forgotten' ruling in the... facebook manipulated users emotions in secret google's self-driving car prototype: no steering wheel, no steering wheel
			Entertain	captain america: the winter soldier 'sets april record with $96.2m in... 'game of thrones' season 4 episode 4 recap: 'the lion and the rose' taylor swift's' music music 'is a paid for $50 millione
			Health	google's self-driving cars are mastering city streets: study stephen colbert to replace david letterman on 'the late show' neil patrick harris poses for a rolling stone 'in the face

Table 9: Sentence samples generated by conditioning on sentiment attribute. The models are trained on News Titles. The sentences are random selected with the help of RANDOM.SHUFFLE().