

---

# Latent Diffusion Energy-Based Model for Interpretable Text Modeling

---

Peiyu Yu<sup>1,2</sup> Sirui Xie<sup>1</sup> Xiaojian Ma<sup>1,2</sup> Baoxiong Jia<sup>1,2</sup> Bo Pang<sup>3</sup>  
Ruiqi Gao<sup>4</sup> Yixin Zhu<sup>5,6</sup> Song-Chun Zhu<sup>1,2,5,6,7,8</sup> Ying Nian Wu<sup>7</sup>

## Abstract

Latent space Energy-Based Models (EBMs), also known as energy-based priors, have drawn growing interests in generative modeling. Fueled by its flexibility in the formulation and strong modeling power of the latent space, recent works built upon it have made interesting attempts aiming at the interpretability of text modeling. However, latent space EBMs also inherit some flaws from EBMs in data space; the degenerate MCMC sampling quality in practice can lead to poor generation quality and instability in training, especially on data with complex latent structures. Inspired by the recent efforts that leverage diffusion recovery likelihood learning as a cure for the sampling issue, we introduce a novel symbiosis between the diffusion models and latent space EBMs in a variational learning framework, coined as the *latent diffusion energy-based model*. We develop a geometric clustering-based regularization jointly with the information bottleneck to further improve the quality of the learned latent space. Experiments on several challenging tasks demonstrate the superior performance of our model on interpretable text modeling over strong counterparts.

## 1. Introduction

Text modeling has achieved impressive progress with the fast development of neural generative models (Serban et al., 2016; Li et al., 2017a; Zhao et al., 2017; Gupta et al., 2018; Zhao et al., 2018a). It allows near human-level text gener-

Code repo and data: <https://github.com/Peiyu98/LDEBM>.

<sup>1</sup>Department of Computer Science, UCLA, USA <sup>2</sup>Beijing Institute for General Artificial Intelligence, China <sup>3</sup>Salesforce Research, USA <sup>4</sup>Google Brain, USA <sup>5</sup>Institute for Artificial Intelligence, Peking University, China <sup>6</sup>School of Artificial Intelligence, Peking University, China <sup>7</sup>Department of Statistics, UCLA, USA <sup>8</sup>Department of Automation, Tsinghua University, China.

Correspondence to: Peiyu Yu <yupeiyu98@cs.ucla.edu>.

*Proceedings of the 39<sup>th</sup> International Conference on Machine Learning*, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

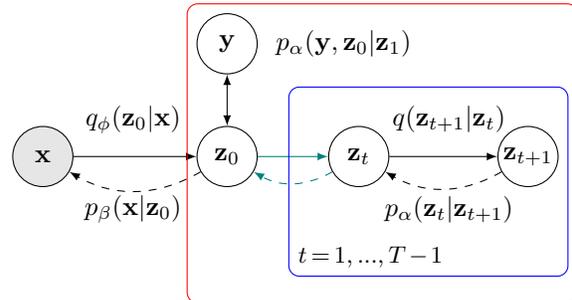


Figure 1. Graphical illustration of the latent diffusion process.

We construct the forward and reverse diffusion processes in the latent space. The symbolic one-hot vector is coupled with the initial latent vector  $\mathbf{z}_0$ . The latent and diffused latent variables are highlighted by the red and blue plates, respectively. The cyan arrows indicate that  $\mathbf{z}_0$  is connected with only  $\mathbf{z}_1$ . We learn a sequence of EBMs to model the reverse diffusion process  $p_\alpha(\mathbf{z}_t|\mathbf{z}_{t+1})$ .

ation quality and also leads to a wide range of real-world applications such as dialog system (Young et al., 2013) and machine translation (Brown et al., 1993). Although the quality of generation (*e.g.*, fluency and diversity) is the primary concern of most work, interpretability of the generation process has drawn much attention recently. Among the existing frameworks, the Deep Latent Variable Model (DLVM) is especially suitable for the task, as the learned latent space could capture high-level structures with semantic meanings like topics (Wang et al., 2019) and dialog actions (Zhao et al., 2018b); such latent space could further enable more interpretable text modeling, featuring unsupervised text attributes discovery (Wen et al., 2017), conditional and controllable text generation (Fang et al., 2019; Shi et al., 2020), and semi-supervised text classification (Pang & Wu, 2021).

In essence, DLVM summarizes the observed sample (*e.g.*, a piece of text) into inferred latent variables. Earlier text-modeling methods with DLVM mostly follow the formulation of Variational Auto-Encoder (VAE) (Kingma & Welling, 2013; Rezende et al., 2014; Bowman et al., 2016), which assumes a continuous latent space. More recently, Zhao et al. (2018b) explore the possibility of using a discrete latent space to capture dialog actions; Shi et al. (2020) propose to use VAE with the mixture of Gaussians as the prior, demonstrating promising interpretability of dialog utterance generation. To further improve the expressivity of the latent

space, Pang & Wu (2021) leverage the flexibility of *energy-based prior* (Pang et al., 2020a) and learn a structured latent space for interpretable text generation and classification. Specifically, they propose a symbol-vector coupling prior model. The continuous latent variables are coupled with discrete one-hot symbol variables, allowing better discrete structure induction without sacrificing the generation quality offered by the continuous latent space. However, similar to learning an EBM in data space, the learning of energy-based prior requires Markov Chain Monte Carlo (MCMC) sampling, whose quality can degenerate in practice (Grathwohl et al., 2019; Nijkamp et al., 2019; 2020; Gao et al., 2020), especially on data with complex latent structures; it often leads to instability during training. As we demonstrate empirically in Sec. 4.1, this phenomenon is particularly concerning when adopting the variational learning scheme to update model parameters.

To remedy this MCMC sampling issue, we may take a look at the endeavor of EBM learning in general. Among the recent efforts, methods drawn inspiration from the diffusion probabilistic models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song & Ermon, 2020; Song et al., 2020) have demonstrated superior results. In particular, Gao et al. (2020) propose a diffusion recovery likelihood method to learn and sample from a sequence of EBMs defined on increasingly noisy versions of a dataset; the models are trained by optimizing conditional likelihoods, which are more tractable than the marginal likelihood. It greatly mitigates the burden of sampling during training. A natural question thus emerges: *Can we leverage the methodology of diffusion models to address the learning issue of energy-based prior?*

In this work, we make the first attempt to address the learning issue of energy-based prior through leveraging diffusion models in the latent space, with a focus on interpretable text modeling. We first unveil the non-trivial symbiosis between latent-space EBMs and diffusion models. Specifically, we focus on the symbol-vector coupling prior; we construct a flexible process that restores the hidden structure in text data by noise-level-aware sampling from a learned sequence of conditional EBMs in the latent space. A variational learning framework is then derived from it. We further employ a geometric clustering-based regularization that complements the symbol-inducing information bottleneck to improve the quality of learned latent space. We term the resulting model Latent Diffusion Energy-Based Model (LDEBM). Compared to Gao et al. (2020), which deals with EBMs in the data space, LDEBM is directly applicable to text data with or without labels; it extracts interpretable latent structures that benefit potential downstream tasks such as semi-supervised classification. Although there are methods using diffusion models in the latent space, some of which have achieved very impressive image generation results, e.g., Vahdat et al. (2021), few of them to our knowl-

edge have explored (unsupervised) symbol induction in the latent space especially on text data. In addition, our method can be trained from scratch and form a well-structured latent space without pretraining, as required by concurrent works on image modeling such as Vahdat et al. (2021) and Nie et al. (2021). In our experiments on generative modeling and interpretable text modeling, LDEBM largely outperforms strong counterparts in terms of both generation quality and interpretability of the learned latent space.

**Contributions** (1) We introduce a novel symbiosis of the latent space EBM and diffusion model in a variational learning framework; the model can be trained from scratch, is directly applicable to text data with or without labels, and shows superior sampling quality. (2) We develop a geometric clustering-based regularization jointly with the information bottleneck that tackles the mode-collapse problem in variational learning of the latent space EBM. (3) Our experiments demonstrate that the proposed model learns a well-structured latent space and delivers strong results on interpretable text modeling.

## 2. Preliminaries: Symbol-Vector Coupling EBM

We assume that for an observed high-dimensional sample  $\mathbf{x} \in \mathbb{R}^D$ , there exists  $\mathbf{z} \in \mathbb{R}^d$  as its compact continuous latent variables. We assume that  $\mathbf{y}$  is the symbolic one-hot vector indicating one of  $K$  categories that  $\mathbf{z}$  belongs to. The complete-data distribution is  $p_\theta(\mathbf{y}, \mathbf{z}, \mathbf{x}) = p_\alpha(\mathbf{y}, \mathbf{z})p_\beta(\mathbf{x}|\mathbf{z})$ , where  $p_\alpha(\mathbf{y}, \mathbf{z})$  is the joint prior model with parameters  $\alpha$ , and  $p_\beta(\mathbf{x}|\mathbf{z})$  is the top-down generation model with parameters  $\beta$ ; henceforth, we use  $\theta = (\alpha, \beta)$  to summarize the parameters. Given  $\mathbf{z}$ ,  $\mathbf{y}$  and  $\mathbf{x}$  are independent; i.e.,  $\mathbf{z}$  is sufficient for  $\mathbf{y}$  in this model.

Pang & Wu (2021) propose to formulate the joint prior model,  $p_\alpha(\mathbf{y}, \mathbf{z})$ , as an EBM,

$$p_\alpha(\mathbf{y}, \mathbf{z}) = \frac{1}{Z_\alpha} \exp(\langle \mathbf{y}, f_\alpha(\mathbf{z}) \rangle) p_0(\mathbf{z}), \quad (1)$$

where  $p_0(\mathbf{z})$  is a reference distribution, assumed to be the non-informative prior (e.g., isotropic Gaussian or uniform) of the conventional generation model,  $f_\alpha(\mathbf{z}) \in \mathbb{R}^K$  is parameterized by a small multi-layer perceptron, and  $Z_\alpha$  is the normalizing constant or partition function. The energy term  $\langle \mathbf{y}, f_\alpha(\mathbf{z}) \rangle$  in Eq. (1) forms an associative memory that couples the symbol  $\mathbf{y}$  and the dense vector  $\mathbf{z}$ . Given  $\mathbf{z}$ ,

$$p_\alpha(\mathbf{y}|\mathbf{z}) \propto \exp(\langle \mathbf{y}, f_\alpha(\mathbf{z}) \rangle) \quad (2)$$

becomes a softmax classifier, where  $f_\alpha(\mathbf{z})$  provides the logit scores for the  $K$  categories. Marginally, we have

$$p_\alpha(\mathbf{z}) = \frac{1}{Z_\alpha} \exp(F_\alpha(\mathbf{z})) p_0(\mathbf{z}), \quad (3)$$

where the marginal energy term is in a log-sum-exponential form,  $F_\alpha(\mathbf{z}) = \log \sum_{\mathbf{y}} \exp(\langle \mathbf{y}, f_\alpha(\mathbf{z}) \rangle)$ . It is shown that the coupling between  $\mathbf{z}$  and  $\mathbf{y}$  enables a symbol-aware continuous vector computation during prior and posterior sampling, which helps to induce a structural latent space (Pang & Wu, 2021). Finally, the prior model  $p_\alpha(\mathbf{y}, \mathbf{z})$  stands on a generation model  $p_\beta(\mathbf{x}|\mathbf{z})$ . In text modeling, let  $\mathbf{x} = (\mathbf{x}^{(t)}, t = 1, \dots, T)$  be a sentence, where  $\mathbf{x}^{(t)}$  is the  $t$ -th token.  $p_\beta(\mathbf{x}|\mathbf{z})$  can be defined as a conditional autoregressive model,  $p_\beta(\mathbf{x}|\mathbf{z}) = \prod_{t=1}^T p_\beta(\mathbf{x}^{(t)}|\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(t-1)}, \mathbf{z})$ . The complete model  $p_\theta(\mathbf{y}, \mathbf{z}, \mathbf{x})$  with the energy-based prior  $p_\alpha(\mathbf{y}, \mathbf{z})$  and the generation model  $p_\beta(\mathbf{x}|\mathbf{z})$  is termed as Symbol-Vector Coupling Energy-Based Model (SVEBM).

In principle, a SVEBM can be learned through maximizing the log-likelihood function, where the learning gradient is  $\nabla_\theta \log p_\theta(\mathbf{x}) = \mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{x})}[\nabla_\theta(\log p_\alpha(\mathbf{z}) + \log p_\beta(\mathbf{x}|\mathbf{z}))]$ . To estimate the expectation, one may sample from the prior  $p_\alpha(\mathbf{z})$  and the posterior  $p_\theta(\mathbf{z}|\mathbf{x})$  with Langevin dynamics (Welling & Teh, 2011). Since  $f_\alpha$  is a small network, prior sampling is particularly affordable. In comparison, the posterior sampling can be more expensive as it requires back-propagating through the generation network. One promising solution is to follow the variational learning scheme (Kingma & Welling, 2013) that amortizes the posterior sampling from  $p_\theta(\mathbf{z}|\mathbf{x})$  by an inference network  $q_\phi(\mathbf{z}|\mathbf{x})$ ; MCMC-based sampling can be used for prior samples.

### 3. Latent Diffusion Energy-Based Model

#### 3.1. A Symbiosis between SVEBM and Diffusion Model

Contrasting to the vanilla sampling process of the latent variables in SVEBM, LDEBM follows the philosophy of diffusion probabilistic models (Sohl-Dickstein et al., 2015); it assumes a sequence of perturbed samples,  $\mathbf{z}_0, \mathbf{z}_1, \dots, \mathbf{z}_T$ , to construct a flexible process that restores the structure in data. First, we define the forward diffusion process that systematically and gradually destroys structure in a data distribution:  $\mathbf{z}_0 \sim q_\phi(\mathbf{z}_0|\mathbf{x}); \mathbf{z}_{t+1} = \sqrt{1 - \sigma_{t+1}^2} \mathbf{z}_t + \sigma_{t+1} \boldsymbol{\epsilon}_{t+1}$ , where  $t = 0, 1, \dots, T-1$  and  $\boldsymbol{\epsilon}_t$  is the zero-mean standard Gaussian noise. The scaling factor  $\sqrt{1 - \sigma_{t+1}^2}$  ensures that the sequence is a spherical interpolation between the posterior sample and the Gaussian white noise. The forward trajectory and the Markov transition between each perturbed samples  $\mathbf{z}_1, \dots, \mathbf{z}_T$  are thus

$$q_\phi(\mathbf{z}_{0:T}|\mathbf{x}) = q_\phi(\mathbf{z}_0|\mathbf{x}) \prod_{t=0}^{T-1} q(\mathbf{z}_{t+1}|\mathbf{z}_t); \quad (4)$$

$$q(\mathbf{z}_{t+1}|\mathbf{z}_t) = \mathcal{N}(\mathbf{z}_{t+1}; \sqrt{1 - \sigma_{t+1}^2} \mathbf{z}_t, \sigma_{t+1}^2 \mathbf{I}).$$

Our goal is to learn the generative distribution that describes the same trajectory but in reverse. Inspired by Gao et al.

(2020), we start by constructing a sequence of *marginal* EBMs at each diffusion step in the latent space. The *conditional* EBMs aiming at recovering  $\mathbf{z}_0$  from noisy inputs then follow as (see the derivation in Appx. A.1):

$$p_\alpha(\tilde{\mathbf{z}}_t|\mathbf{z}_{t+1}) = \frac{1}{\tilde{Z}_{\alpha,t}(\mathbf{z}_{t+1})} \exp\left(F_\alpha(\tilde{\mathbf{z}}_t, t) - \frac{1}{2\sigma_{t+1}^2} \|\tilde{\mathbf{z}}_t - \mathbf{z}_{t+1}\|^2\right), \quad (5)$$

where  $t = 0, 1, \dots, T-2$ . We denote  $\tilde{\mathbf{z}}_t = \sqrt{1 - \sigma_{t+1}^2} \mathbf{z}_t$  for brevity.  $F_\alpha(\tilde{\mathbf{z}}_t, t)$  is the neural network that parameterizes the energy function at each diffusion step, and  $\tilde{Z}_{\alpha,t}(\mathbf{z}_{t+1}) = \int \exp(F_\alpha(\tilde{\mathbf{z}}_t, t) - \frac{1}{2\sigma_{t+1}^2} \|\tilde{\mathbf{z}}_t - \mathbf{z}_{t+1}\|^2) d\tilde{\mathbf{z}}_t$  is the partition function of each conditional EBM. For  $t = T-1$ ,  $p_\alpha(\tilde{\mathbf{z}}_t|\mathbf{z}_{t+1}) = \frac{1}{\tilde{Z}_{\alpha,t}} \exp(F_\alpha(\tilde{\mathbf{z}}_t, t) - \frac{1}{2\sigma_{t+1}^2} \|\tilde{\mathbf{z}}_t\|^2)$  since the diffused samples at time step  $T$  should be close to Gaussian white noise; the distribution of  $\tilde{\mathbf{z}}_{T-1}$  can thus be exponentially tilting of a zero-mean Gaussian distribution.

Eq. (5) shares the idea of denoising generative modeling (Bengio et al., 2013), where a denoising autoencoder is trained by maximizing the conditional probabilities of the observed samples given their noisy versions. Compared to the vanilla definition (see Eq. (3)), the noise-level-aware quadratic term constrains the energy landscape to be localized around the noisy sample; this makes the latent space much less multi-modal and easier to sample from. To be specific, Gao et al. (2020) show that  $p_\alpha(\tilde{\mathbf{z}}_t|\mathbf{z}_{t+1})$  is approximately a single-mode Gaussian distribution when  $\sigma$  is sufficiently small; it greatly reduces the burden of MCMC sampling. After sampling  $\tilde{\mathbf{z}}_t$  from the model, we can easily obtain  $\mathbf{z}_t = \tilde{\mathbf{z}}_t / \sqrt{1 - \sigma_{t+1}^2}$ .

Next, we show that the forward and reverse process in the latent space can be naturally integrated into the variational learning scheme to amortize the time-consuming posterior sampling. Similar to VAE, the ELBO in SVEBM is

$$\text{ELBO}_{\theta, \phi} = \log p_\theta(\mathbf{x}) - \mathbb{D}_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z}|\mathbf{x})) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\beta(\mathbf{x}|\mathbf{z})] - \mathbb{D}_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\alpha(\mathbf{z})), \quad (6)$$

where  $\mathbb{D}_{\text{KL}}$  denotes the Kullback-Leibler divergence. Since we now consider the full trajectory of the perturbed samples, in LDEBM we may optimize

$$\text{ELBO}_{\text{Diff}, \theta, \phi} = \mathbb{E}_{q_\phi(\mathbf{z}_0|\mathbf{x})} [\log p_\beta(\mathbf{x}|\mathbf{z}_0) - \log q_\phi(\mathbf{z}_0|\mathbf{x})] + \mathbb{E}_{q_\phi(\mathbf{z}_0|\mathbf{x}), q(\mathbf{z}_{1:T}|\mathbf{z}_0)} \left[ \log \frac{p_\alpha(\mathbf{z}_{0:T})}{q(\mathbf{z}_{1:T}|\mathbf{z}_0)} \right], \quad (7)$$

which is a valid ELBO by applying Jensen's inequality to Eq. (6). The joint training of inference, prior and generation models can be largely reduced to finding the agreement of the forward and reverse Markov transitions defined by  $q_\phi$  and  $p_\theta$ , respectively. Please refer to Appx. A.2 for more detailed derivations and discussions.

Finally, we show how to introduce the symbolic one-hot vector  $\mathbf{y}$  into our formulation. We assume a complete data distribution that considers the full trajectory of the perturbed latent variables,  $p_\theta(\mathbf{y}, \mathbf{z}_{0:T}, \mathbf{x})$ . Among several possibilities for coupling the symbolic vector  $\mathbf{y}$  with the latent variables, two major options arise: We can couple the symbol with the whole trajectory, *i.e.*,  $p_\theta(\mathbf{y}, \mathbf{z}_{0:T}, \mathbf{x}) = p_\alpha(\mathbf{y}, \mathbf{z}_{0:T})p_\beta(\mathbf{x}|\mathbf{z}_{0:T})$ ; or we can couple the symbol with only the clean posterior sample  $\mathbf{z}_0$ , *i.e.*,  $p_\theta(\mathbf{y}, \mathbf{z}_{0:T}, \mathbf{x}) = p(\mathbf{z}_T)p_\alpha(\mathbf{y}, \mathbf{z}_0|\mathbf{z}_1) \prod_{t=1}^{T-1} p_\alpha(\mathbf{z}_t|\mathbf{z}_{t+1})p_\beta(\mathbf{x}|\mathbf{z}_0)$ . We prefer the latter one, since it is sufficient to model the reverse Markovian transition, while enabling a simpler and more efficient training scheme following [Ho et al. \(2020\)](#) (see [Sec. 3.4](#)). Of note, coupling only  $\mathbf{z}_0$  to  $\mathbf{y}$  means that we condition only the final reverse diffusion step  $[\mathbf{z}_0|\mathbf{z}_1]$  on  $\mathbf{y}$  when performing controllable generation. This could be a bit counter-intuitive as no label information is injected in previous reverse steps. Theoretically,  $\mathbf{y}$  and  $\mathbf{z}_{1:T}$  are independent given  $\mathbf{z}_0$  in our formulation; however, we empirically observe that  $\mathbf{y}$  and  $\mathbf{z}_t$  for  $t > 0$  are nearly independent even marginally, after we integrating out  $\mathbf{z}_{0:t-1}$  in our model. In other words,  $p_\alpha(\mathbf{y}|\mathbf{z}_t)$ ,  $t > 0$  are in general non-informative since adding noise in the latent space could be much more corrupting than adding noise in the data space. The model learns to enjoy the less multi-modal energy landscape in previous reverse steps; it then seeks the given mode only in the most informative final reverse step. Specifically, we achieve this coupling by similarly defining  $p_\alpha(\mathbf{y}, \mathbf{z}_0|\mathbf{z}_1)$  as in [Eq. \(1\)](#) and using the log-sum-exponential form for learning as in [Eq. \(3\)](#). Please refer to [Fig. 1](#) for a graphical illustration of our model and [Appx. A.3](#) and [B.3](#) for more details and discussions.

### 3.2. Information Bottleneck

To learn the symbolic vector  $\mathbf{y}$ , we may consider adopting the Information Bottleneck (IB) principle ([Tishby et al., 2000](#)), an appealing approach for inducing symbolic representations. In this section, we re-interpret the above ELBO as a cooperative learning objective, defined as the divergence between two joint distributions; we then show how this formulation helps to incorporate the IB-based regularization into LDEBM in a principled manner.

As shown in [Han et al. \(2019\)](#), the variational learning scheme can be regarded as performing alternating projection between two joint distributions,  $Q_\phi$  and  $P_\theta$ . In our modeling, we have:  $Q_\phi(\mathbf{x}, \mathbf{z}_{0:T}) = q_{\text{data}}(\mathbf{x})q_\phi(\mathbf{z}_{0:T}|\mathbf{x})$ , and  $P_\theta(\mathbf{x}, \mathbf{z}_{0:T}) = p(\mathbf{z}_T) \prod_{t=0}^{T-1} p_\alpha(\mathbf{z}_t|\mathbf{z}_{t+1})p_\beta(\mathbf{x}|\mathbf{z}_0)$ ; we use  $q_{\text{data}}(\mathbf{x})$  to denote the data distribution of  $\mathbf{x}$  for notation consistency. Maximizing  $\mathbb{E}_{q_{\text{data}}(\mathbf{x})}[\text{ELBO}_{\text{Diff}, \theta, \phi}(\mathbf{x})]$  over  $(\theta, \phi)$  is equivalent to minimizing the following divergence:

$$\begin{aligned} \mathbb{D}_{\text{KL}}(Q_\phi \| P_\theta) &= \mathbb{D}_{\text{KL}}(q_{\text{data}}(\mathbf{x}) \| p_\theta(\mathbf{x})) \\ &+ \mathbb{E}_{q_{\text{data}}(\mathbf{x})}[\mathbb{D}_{\text{KL}}(q_\phi(\mathbf{z}_{0:T}|\mathbf{x}) \| p_\theta(\mathbf{z}_{0:T}|\mathbf{x}))], \end{aligned} \quad (8)$$

since  $\mathcal{H}(\mathbf{x}) = -\mathbb{E}_{q_{\text{data}}(\mathbf{x})}[\log q_{\text{data}}(\mathbf{x})]$ , *i.e.*, the entropy of data distribution is fixed. Minimizing the KL-divergence  $\min_\theta \min_\phi \mathbb{D}_{\text{KL}}(Q_\phi \| P_\theta)$  defines a cooperative game, with the dynamics that  $q_\phi$  and  $p_\theta$  run towards each other.

Since the initial posterior sample  $\mathbf{z}_0$  is coupled with the symbolic vector  $\mathbf{y}$ , it should be the most informative latent variable for inducing the discrete symbol. We can therefore plug in [Eq. \(8\)](#) with a mutual information term between  $\mathbf{z}_0$  and  $\mathbf{y}$ :  $\mathcal{I}(\mathbf{z}_0, \mathbf{y}) = \mathcal{H}(\mathbf{y}) - \mathcal{H}(\mathbf{y}|\mathbf{z}_0)$ , which essentially incorporates the IB as we show below. Given the distribution  $Q_\phi(\mathbf{x}, \mathbf{z}_{0:T})$ , we can first define the marginal distribution of  $\mathbf{z}_0$  as the aggregated posterior by integrating out  $\mathbf{z}_{1:T}$ :  $q_\phi(\mathbf{z}_0) = \mathbb{E}_{q_{\text{data}}(\mathbf{x})}[q_\phi(\mathbf{z}_0|\mathbf{x})]$ . The entropy of  $\mathbf{z}_0$  and conditional entropy of  $\mathbf{z}_0$  on  $\mathbf{x}$  then follow as  $\mathcal{H}(\mathbf{z}_0)$  and  $\mathcal{H}(\mathbf{z}_0|\mathbf{x})$ , respectively. Taken together, the KL-Divergence with  $\lambda \mathcal{I}(\mathbf{z}_0, \mathbf{y})$  can therefore be parsed as

$$\begin{aligned} \mathcal{L} &= \mathbb{D}_{\text{KL}}(Q_\phi \| P_\theta) - \lambda \mathcal{I}(\mathbf{z}_0, \mathbf{y}) \\ &= \mathcal{C} + \mathcal{L}_{\text{RC}} + \mathcal{L}_{\text{EBM}} + \mathcal{L}_{\text{IB}}, \end{aligned} \quad (9)$$

where  $\mathcal{C} = -\mathcal{H}(\mathbf{x}) + \sum_{t=0}^{T-1} \mathcal{H}(\mathbf{z}_{t+1}|\mathbf{z}_t)$  does not involve learnable parameters,  $\mathcal{L}_{\text{RC}} = -\mathbb{E}_{Q_\phi}[\log p_\beta(\mathbf{x}|\mathbf{z}_0)]$  is the reconstruction loss,  $\mathcal{L}_{\text{EBM}} = \mathbb{D}_{\text{KL}}(q_\phi(\mathbf{z}_0) \| p_\alpha(\mathbf{z}_{0:T}))$  corresponds with learning latent space models, and  $\mathcal{L}_{\text{IB}} = \mathcal{I}(\mathbf{x}, \mathbf{z}_0) - \lambda \mathcal{I}(\mathbf{z}_0, \mathbf{y})$  is the IB, where  $\mathcal{I}(\mathbf{x}, \mathbf{z}_0) = \mathcal{H}(\mathbf{z}_0) - \mathcal{H}(\mathbf{z}_0|\mathbf{x})$  is the mutual information between  $\mathbf{x}$  and  $\mathbf{z}_0$  under  $Q_\phi$ ;  $\lambda \geq 0$  controls the expressivity of  $\mathbf{z}_0$  to  $\mathbf{y}$ . Please refer to [Appx. A.4](#) for more details.

### 3.3. Geometric Clustering Anchors the Modes

As shown in the previous section, IB provides an elegant solution for inducing the symbolic vector  $\mathbf{y}$ . In this section, we further introduce an approach that facilitates the emergence of  $\mathbf{y}$  from a geometric perspective. To induce a latent space with interpretable structures, ideally, the location of data points in the latent space encodes their semantic meaning, *i.e.*, it indicates the semantic class; semantically similar points should be placed closer and produce the same symbolic vector  $\mathbf{y}$ . This resembles geometric clustering algorithms: Labels of data points are assigned based on their geometric (typically Euclidean) distance from each other. Below, we show how to realize this intuition in LDEBM.

Let us consider the joint distribution  $p_\theta(\mathbf{x}, \mathbf{y})$ . We can decompose its log-likelihood into  $\log p_\theta(\mathbf{x}, \mathbf{y}) = \log p_\theta(\mathbf{x}) + \log p_\theta(\mathbf{y}|\mathbf{x})$  as in [Grathwohl et al. \(2019\)](#), where  $\log p_\theta(\mathbf{x})$  is substituted with the ELBO derived in [Sec. 3.1](#).  $p_\theta(\mathbf{y}|\mathbf{x})$  is the classification model in the latent space:  $p_\theta(\mathbf{y}|\mathbf{x}) \approx \mathbb{E}_{q_\phi(\mathbf{z}_0|\mathbf{x})}[p_\alpha(\mathbf{y}|\mathbf{z}_0)]$ .  $p_\alpha(\mathbf{y}|\mathbf{z}_0)$  is the softmax classifier of  $\mathbf{y}$  based on  $\mathbf{z}_0$  similarly as in [Eq. \(2\)](#), detailed in [Appx. A.3](#). Therefore, we can encode the semantic information from the label  $\mathbf{y}$  into  $\mathbf{z}_0$  through learning the classifier  $p_\alpha(\mathbf{y}|\mathbf{z}_0)$ . In case there is full or partial access to the ground-truth semantic class labels, we could directly utilize these labels

to supervise the classifier, jointly with the existing ELBO objective. When no label is provided, we generate pseudo label  $\hat{\mathbf{y}}$  by clustering  $\mathbf{z}_0$ , which optimizes  $\mathbb{E}_{\mathbf{y}} \log p_{\theta}(\mathbf{x}, \mathbf{y})$  instead;  $\mathbb{E}_{\mathbf{y}}$  is defined by the clustering algorithm. It is akin to the EM algorithm, where geometric clustering serves as a hard-decision E-step to help induce  $\mathbf{y}$ . In practice, we employ K-means to cluster  $\mathbf{z}_0$ . In Sec. 4.1, we empirically show that this strategy learns a better latent space and significantly alleviates the mode-collapse problem.

### 3.4. Algorithms and Implementation

#### Algorithm 1 Learning algorithm.

**input:** initial parameters  $(\alpha, \beta, \phi)$ , learning rate  $\eta$ , observed unlabeled examples  $\{\mathbf{x}^{(i)}\}_{i=1}^M$ , observed labeled examples  $\{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=M+1}^{M+N}$  (alternative, needed in controllable generation or semi-supervised learning).

**repeat**

**posterior sampling:** For each  $\mathbf{x}^{(i)}$ , sample  $\mathbf{z}_0^{(i)} \sim q_{\phi}(\mathbf{z}_0 | \mathbf{x}^{(i)})$  using inference network.

**prior sampling:** For each  $\mathbf{z}_0^{(i)}$ , sample diffusion step  $t$  from  $\text{Unif}(\{0, \dots, T-1\})$ , and the perturbed pair  $(\tilde{\mathbf{z}}_t^{(i)}, \mathbf{z}_{t+1}^{(i)})$  following Eq. (4). Set  $\tilde{\mathbf{z}}_t^{(i)}$  as the positive sample  $\tilde{\mathbf{z}}_t^{(i)+}$ . Initialize the MCMC using  $\mathbf{z}_{t+1}^{(i)}$  and update by Eq. (12) for  $K$  steps to obtain  $\tilde{\mathbf{z}}_t^{(i)-}$ .

**learning prior model:** Update  $\alpha$  with  $\eta(\sum_i [\nabla_{\alpha} F_{\alpha}(\tilde{\mathbf{z}}_t^{(i)+}, t) - \nabla_{\alpha} F_{\alpha}(\tilde{\mathbf{z}}_t^{(i)-}, t)] - \nabla_{\alpha} \mathcal{I})$ .

**learning inference and generation models:**

Update  $\beta$  and  $\phi$  with Eq. (11) and  $\nabla_{\phi} \mathcal{I}$ .

**if** labeled data  $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$  is available **then**

**update**  $\gamma = (\alpha, \phi)$  **using**  $\mathbf{y}^{(i)}$ :

Learning gradient  $\eta \sum_i \nabla_{\gamma} \log p_{\alpha t}(\mathbf{y}^{(i)} | \mathbf{z}_0^{(i)})$  is provided by ground-truth label.

**else if** only unlabeled data is available **then**

**update**  $\gamma = (\alpha, \phi)$  **using pseudo-label**  $\hat{\mathbf{y}}^{(i)}$ :

Geometric clustering generates  $\hat{\mathbf{y}}^{(i)}$  for each  $\mathbf{x}^{(i)}$ .  $\eta \sum_i \nabla_{\gamma} \log p_{\alpha t}(\hat{\mathbf{y}}^{(i)} | \mathbf{z}_0^{(i)})$ , i.e., the gradient comes from pseudo-label generated by geometric clustering.

**end if**

**until** converged.

**Training and sampling algorithms** For learning the prior model, we have for each  $t = 0, 1, \dots, T-1$ :

$$\begin{aligned} \nabla_{\alpha} \text{ELBO}_t &= \mathbb{E}_{q_{\phi}(\tilde{\mathbf{z}}_t, \mathbf{z}_0 | \mathbf{x})} [\nabla_{\alpha} F_{\alpha}(\tilde{\mathbf{z}}_t, t)] \\ &\quad - \mathbb{E}_{q_{\phi}(\mathbf{z}_{t+1}, \mathbf{z}_0 | \mathbf{x}), p_{\alpha}(\tilde{\mathbf{z}}_t | \mathbf{z}_{t+1})} [\nabla_{\alpha} F_{\alpha}(\tilde{\mathbf{z}}_t, t)]. \end{aligned} \quad (10)$$

Let  $\psi = \{\beta, \phi\}$  collect the parameters of the inference (encoder) and generation (decoder) models.

$$\begin{aligned} \nabla_{\psi} \text{ELBO} &= \nabla_{\psi} \mathbb{E}_{q_{\phi}(\mathbf{z}_0 | \mathbf{x})} [\log p_{\beta}(\mathbf{x} | \mathbf{z}_0) - \log q_{\phi}(\mathbf{z}_0 | \mathbf{x})] \\ &\quad - \nabla_{\phi} \mathbb{E}_{q_{\phi}(\mathbf{z}_0, T | \mathbf{x})} \left[ \log p(\mathbf{z}_T) + \sum_{t=0}^{T-1} \log p_{\alpha}(\mathbf{z}_t | \mathbf{z}_{t+1}) \right]. \end{aligned} \quad (11)$$

Recall that we denote  $\tilde{\mathbf{z}}_t = \sqrt{1 - \sigma_{t+1}^2} \mathbf{z}_t$ .  $\mathbb{E}_{p_{\alpha}(\tilde{\mathbf{z}}_t | \mathbf{z}_{t+1})}$  is approximated by MCMC samples from the prior.  $\mathbb{E}_{q_{\phi}(\mathbf{z}_0 | \mathbf{x})}$  is approximated by samples from the inference network. We also add the gradient from  $\mathcal{I}(\mathbf{z}_0, \mathbf{y})$ , denoted as  $\nabla \mathcal{I}$ , to Eqs. (10) and (11) during training to incorporate IB. Please see Appx. A.5 for detailed derivations.

Note that the expectation in Eq. (10) requires MCMC sampling (e.g., Langevin dynamics (Welling & Teh, 2011)) of the prior model. For a target distribution  $\pi(\tilde{\mathbf{z}})$ , the dynamics iterates  $\tilde{\mathbf{z}}^{k+1} = \tilde{\mathbf{z}}^k + \frac{s^2}{2} \nabla_{\tilde{\mathbf{z}}} \log \pi(\tilde{\mathbf{z}}^k) + s \epsilon^k$ , where  $k$  indexes the iteration of the dynamics,  $s$  is a small step size, and  $\epsilon^k \sim \mathcal{N}(0, \mathbf{I})$  is the Gaussian noise. In this work, we follow the heuristics in Gao et al. (2020) and set the step size  $s_t = b \sigma_t c_t$ , where  $b < 1$  is a tuned hyperparameter, and  $c_t = \sqrt{\prod_{i=1}^t \sigma_i / \sigma_1}$  is a scaling factor. Let  $t$  indexes the diffusion step;  $K$  steps of Langevin dynamics thus iterates

$$\begin{aligned} \tilde{\mathbf{z}}_t^{k+1} &= \tilde{\mathbf{z}}_t^k + \frac{b^2 \sigma_t^2 c_t^2}{2} \left( \nabla_{\tilde{\mathbf{z}}} F_{\alpha}(\tilde{\mathbf{z}}_t^k, t) - \frac{1}{\sigma_t^2} (\tilde{\mathbf{z}}_t^k - \mathbf{z}_{t+1}) \right) \\ &\quad + b \sigma_t c_t \epsilon^k. \end{aligned} \quad (12)$$

#### Algorithm 2 Synthesizing algorithm.

**input:**  $\mathbf{z}_T \sim \mathcal{N}(0, \mathbf{I})$

**output:**  $\mathbf{z}_0$

**for**  $t = T-1$  **to**  $t = 0$  **do**

Initialize  $\tilde{\mathbf{z}}_t = \mathbf{z}_{t+1}$ .

**for**  $k = 1$  **to**  $K$  **do**

Update  $\tilde{\mathbf{z}}_t$  using Eq. (12).

**end for**

$\mathbf{z}_t = \tilde{\mathbf{z}}_t / \sqrt{1 - \sigma_{t+1}^2}$

**end for**

In principle, training the model amounts to minimizing the ELBO in Eq. (7), which requires a summation over all the diffusion steps; it involves sampling a full forward trajectory. To make the training simpler and more efficient, following Ho et al. (2020), we randomly choose one diffusion step from the summation to optimize at each training iteration. After training, we initialize the reverse trajectory from Gaussian white noise. The synthesized sample at each step serves to initialize an MCMC that samples from the model of the previous step. The learning and synthesizing algorithms are summarized in Algs. 1 and 2, respectively.

**Implementation** For the K-means algorithm, we use the implementation of Johnson et al. (2019), which explicitly deals with the empty clusters and trivial parameterization problems. To emphasize that the proposed model shows better capability of modeling latent space, we use the same encoder and decoder as Pang & Wu (2021) for all the experiments. We use a shared network  $F_{\alpha}(\tilde{\mathbf{z}}_t, t)$  for each  $t = 0, 1, \dots, T-1$ ;  $T = 6$ ;  $t$  is encoded by sinusoidal position embedding as in Ho et al. (2020), and we set  $\sigma_t^2$  to increase linearly. For Langevin dynamics, we use  $K = 50$

and  $b^2 = 0.002$  throughout the experiments. See Appx. B.1 for network architecture and further training details.

## 4. Experiments

Through a series of experiments, we empirically examine the capability of our model for generative modeling and interpretability on text modeling tasks. Please refer to Appx. B.2 for additional experiment settings and baselines.

### 4.1. Generative Modeling

**2D synthetic data** We first perform experiments of our model on 2D synthetic datasets as a sanity check to validate our assumptions; results are displayed in Fig. 2. The gap between LDEBM and SVEBM is very clear. As mentioned in Sec. 1, for more complex datasets (*e.g.*, datasets with more modes or more complex data structure), SVEBM struggles to capture regularities in the data; the model is prone to collapse, which features an exploding KL-term and poor performance on generation. We provide more results that show the full evolution of these models during training with more discussions in Appx. B.3. In contrast, LDEBM with-

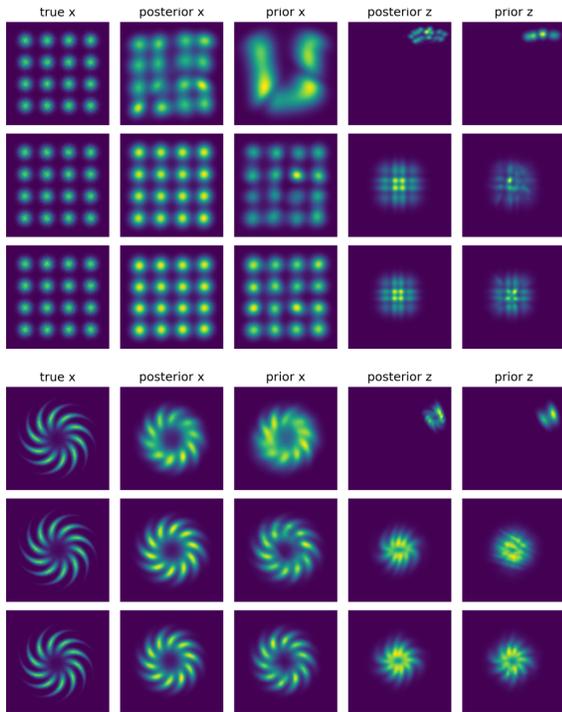


Figure 2. **Evaluation on 2D synthetic data:** a mixture of 16 Gaussians (upper panel) and a 10-arm pinwheel-shaped distribution (lower panel). In each panel, the top, middle, and bottom row display densities learned by SVEBM-IB, our model w/o geometric clustering, and our full model, respectively. In each row, from left to right, it displays the data distribution and the KDEs of:  $\mathbf{x}$  generated by amortized posterior  $\mathbf{z}$  samples,  $\mathbf{x}$  by MCMC sampled prior  $\mathbf{z}$  samples, posterior  $\mathbf{z}$  samples, and prior  $\mathbf{z}$  samples.

out geometric clustering already overcomes this problem, performing relatively well in terms of modeling both *posterior*  $\mathbf{x}$  and *prior*  $\mathbf{x}$ . Although LDEBM without geometric clustering faithfully reconstructs the data and shows significant improvement on generation quality, the generated distribution can be slightly distorted, and some modes are missing. The problem is clearer in the latent space: Mode-collapse occurs in the *prior*  $\mathbf{z}$  distribution, where the latent structure is broken. LDEBM with geometric clustering maintains the number of modes as in the data distribution and induces a highly-structural latent space, echoing our intuition in Sec. 3.3. Fig. 3 shows the structural similarity between data distribution and the learned latent distribution.

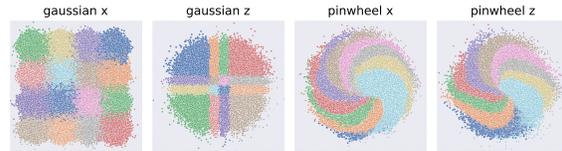


Figure 3. **Visualization of color-coded data points.** We visualize data points and the corresponding inferred latent variables of two 2D synthetic datasets (*gaussian* and *pinwheel*). Data points with different labels are assigned with different colors.

**Language generation** Following previous state-of-the-art competitors (Zhao et al., 2018b; Shi et al., 2020; Pang & Wu, 2021), we evaluate the quality of generation on a real-world text dataset, Penn Treebanks (PTB) (Marcus et al., 1993) as pre-processed by Mikolov et al. (2010). We report four metrics of the generation performance: Reverse Perplexity (rPPL) (Zhao et al., 2018a), BELU (Papineni et al., 2002), Word-Level KL Divergence (wKL), and Negative Log-Likelihood (NLL); Tab. 1 summarizes results.

The proposed model, either w/ or w/o geometric clustering, demonstrates the best performance on reconstruction (highest BLEU) and fitting capacity (lowest NLL) than all baseline models. Moreover, the higher expressivity of our models enables the generation of high-quality sentences. The lowest rPPL indicates that our models can further improve over these strong baselines on fluency and diversity of generated text; the lowest wKL indicates that the word distribution of the generated sentences is the most consistent with that of the original data.

**Sentence completion** Further, we show that the trained model enables text completion on a masked JerichoWorld dataset (Ammanabrolu & Riedl, 2021). We perform conditional sampling in the latent space to complete the masked sentences; please see more details in Appx. B.3 and Tab. 2.

### 4.2. Interpretable Text Modeling

In this section, we move on to evaluate our model on the interpretability of text modeling.

Table 1. Results of language generation on PTB dataset. We highlight our model results in gray color. The best and second-best performances are marked in bold numbers and underlines, respectively; tables henceforth follows this format.

MODEL	rPPL $\downarrow$	BLEU $\uparrow$	wKL $\downarrow$	NLL $\downarrow$
TEST SET	-	100.0	0.14	-
RNN-LM	-	-	-	101.21
AE	730.81	10.88	0.58	-
VAE	686.18	3.12	0.50	100.85
DAE	797.17	3.93	0.58	-
DVAE	744.07	1.56	0.55	101.07
DI-VAE	310.29	4.53	0.24	108.90
SEMI-VAE	494.52	2.71	0.43	100.67
SEMI-VAE + $\mathcal{I}$	260.28	5.08	0.20	107.30
GM-VAE	983.50	2.34	0.72	99.44
GM-VAE + $\mathcal{I}$	287.07	6.26	0.25	103.16
DGM-VAE	257.68	8.17	0.19	104.26
DGM-VAE + $\mathcal{I}$	247.37	8.67	0.18	105.73
SVEBM	180.71	9.54	0.17	95.02
SVEBM-IB	177.59	9.47	0.16	94.68
OURS w/o GC	<u>168.32</u>	<u>11.12</u>	<u>0.07</u>	<b>79.84</b>
OURS	<b>164.57</b>	<b>11.16</b>	<b>0.06</b>	<b>82.38</b>

Table 2. Sentence completion on JerichoWorld dataset. The gray words in the input sentences are masked with  $\langle \text{unk} \rangle$  token.

<b>Input</b>	... A bathroom lies to the south, while a door to the east leads to the living room. On the bed are a driver’s license, some keys and a wallet On the end table is a telephone.
<b>Pred.</b>	... A bathroom lies to the south, while a door to the east leads to the living room. On the bed is a wallet. On the end table are a telephone and some keys.
<b>Input</b>	... All around you the crowd is in a state of pandemonium. The paths of least resistance are up, down and west.
<b>Pred.</b>	... All around you the crowd is in a state of pandemonium. The paths of least resistance are down and east.

**Unsupervised text attributes discovery** First, we examine the efficacy of our model on the unsupervised text attributes discovery task. We assess the model on the Daily Dialog (DD) dataset (Li et al., 2017b), a chat-oriented dataset of 13,118 daily conversations with human-annotated dialog action and emotion labels for the utterances. The interpretability is evaluated through the ability to unsupervisedly capture the utterance attributes of DD. We flatten the dialogues for text modeling and use  $p_{\theta}(y|\mathbf{x})$  to infer the utterance label. In particular, we take the argmax of the classification head as the inferred label. Following Zhao et al. (2018b), we recruit homogeneity to evaluate the consistency between ground-truth action and emotion labels

and those inferred from our model. Tab. 3 displays the results of our model and baselines. It shows that the proposed model outperform other baselines in reconstruction by a large margin and give a much better homogeneity on both the dialog action and emotion. The superior performance of LDEBM equipped with latent space geometric clustering again verifies our intuition in Sec. 3.3.

Table 3. Results of interpretable text modeling on DD. We use mutual information (MI), BLEU, and homogeneity with actions and emotions for evaluation.

MODEL	MI $\uparrow$	BLEU $\uparrow$	Act. $\uparrow$	Emo. $\uparrow$
DI-VAE	1.20	3.05	0.18	0.09
SEMI-VAE	0.03	4.06	0.02	0.08
SEMI-VAE + $\mathcal{I}$	1.21	3.69	0.21	0.14
GM-VAE	0.00	2.03	0.08	0.02
GM-VAE + $\mathcal{I}$	1.41	2.96	0.19	0.09
DGM-VAE	0.53	7.63	0.11	0.09
DGM-VAE + $\mathcal{I}$	1.32	7.39	0.23	0.16
SVEBM	0.01	11.16	0.03	0.01
SVEBM-IB	2.42	10.04	0.59	0.56
OURS w/o GC	2.44	<u>16.72</u>	<u>0.65</u>	<u>0.63</u>
OURS	<b>3.94</b>	<b>28.75</b>	<b>0.74</b>	<b>0.74</b>

**Conditional response generation** Next, we evaluate our model on dialog generation with Stanford Multi-Domain Dialog (SMD) (Eric et al., 2017) and DD datasets. We evaluate the quality of generated responses using BELU and three word-embedding-based topic similarity metrics (Shi et al., 2020): embedding average (Mitchell & Lapata, 2008), embedding extrema (Forgues et al., 2014), and embedding greedy (Rus & Lintean, 2012). Tab. 4 shows that LDEBM has competitive performance compared with SVEBM-IB on SMD and outperforms the strong baselines on all metrics on DD; see qualitative examples in Tabs. 5 and 6.

Table 4. Dialog evaluation results on SMD and DD. Models are assessed using four metrics: BLEU, average, extrema, and greedy word embedding based similarity.

DATA	MODEL	BLEU $\uparrow$	Avg. $\uparrow$	Extr. $\uparrow$	Grdy. $\uparrow$
SMD	DI-VAE	7.06	76.17	43.98	60.92
	DGM + $\mathcal{I}$	10.16	78.93	48.14	64.87
	SVE-IB	<b>12.01</b>	<b>80.88</b>	<u>51.35</u>	<u>67.12</u>
	w/o GC	11.44	<u>80.16</u>	51.26	66.51
	OURS	<u>11.51</u>	<b>80.88</b>	<b>51.57</b>	<b>67.13</b>
DD	DGM + $\mathcal{I}$	2.19	74.73	<u>45.85</u>	<u>64.28</u>
	SVE-IB	<u>2.23</u>	<u>77.37</u>	43.32	63.99
	OURS	<b>3.72</b>	<b>78.89</b>	<b>46.19</b>	<b>65.87</b>

**Sentence sentiment control** Finally, we inspect the capability of our model for controllable generation on Yelp reviews, pre-processed by (Li et al., 2018). The Yelp dataset is of larger scale, containing 180,000 negative reviews and

Table 5. Samples of unsupervisedly discovered action categories and corresponding utterances on SMD.

<b>Action</b>	Request-weather
<b>Utterance</b>	I need to know if it is going to be foggy in Fresno today and tomorrow car. Manhattan, please. Will it be cloudy on Monday? I need current weather data about New York, specifically information about the temperature.
<b>Action</b>	Request-city
<b>Utterance</b>	In what city are you interested? What city would you like to know the weather about? Okay, what city should I look in?

 Table 6. Dialog cases generated by LDEBM given the context. On SMD, we provide the same context but with different  $y$  values to generate each response; actions indicated by  $y$  are listed in parentheses. On DD, LDEBM can well capture the dialog topic; we provide the ground-truth response in each case for reference.

SMD	
<b>Ctx.</b>	<i>User:</i> What gas stations are here? <i>Sys:</i> There is a Chevron.
<b>Ref.</b>	That’s good! Please pick the quickest route to get there and avoid all heavy traffic!
<b>Pred.</b>	(Req.-address) What is the address? (Req.-route) Please set the quickest route to go.
DD	
<b>Ctx.</b>	A: Hi. Have you got a personal computer? B: Certainly. What ’ s the matter? A: I wonder if you often trade with others on the internet.
<b>Ref.</b>	Sure. I often buy things or do business through it without going out to the physical stores.
<b>Pred.</b>	Yes, but I think it is a little different way.

270,000 positive ones. For a controllable generation process, the symbolic vector  $y$  is provided to guide the sampling in latent space; see details in Appx. B.3. Following Pang & Wu (2021), we train the model with sentiment supervision and use the same pre-trained classifier to determine the sentiment of the generated sentence. The pre-trained classifier has an accuracy of 98.5% on the testing data and thus can accurately evaluate the sentiment of given sentences. The quantitative and qualitative results are summarized in Tabs. 7 and 8, respectively. LDEBM generates positive and negative reviews with a nearly saturate accuracy, significantly outperforming all the baselines.

Table 7. Accuracy of sentence attribute control on Yelp.

MODEL	Overall <sup>†</sup>	Positive <sup>†</sup>	Negative <sup>†</sup>
DGM-VAE + $\mathcal{I}$	64.7%	95.3%	34.0%
CGAN	76.8%	94.9%	58.6%
SVEBM-IB	<u>90.1%</u>	<u>95.1%</u>	<u>85.2%</u>
<b>OURS</b>	<b>99.0%</b>	<b>98.8%</b>	<b>99.1%</b>

Table 8. Generated positive and negative reviews on Yelp.

<b>Positive</b>	The food here was very tasty and our server was very attentive. I was very satisfied for my birthday party! Definitely the best Philly Cheesesteaks I’ve ever been. They are the best customer service ever!
<b>Negative</b>	Ugh the staff is so incompetent and rude. It just can’t make it worse. Avoid this company at all costs. Just ruined the experience with a horrible attitude on it.

### 4.3. Semi-supervised Classification

In this experiment, we switch from neural sequence models used in previous experiments to neural document models (Miao et al., 2016; Card et al., 2018); we show our model can be similarly extended to semi-supervised settings as in Pang & Wu (2021) and benefit from the better learned latent space. Our model is evaluated on AGNews (Zhang et al., 2015), a popular benchmark for text classification with 127,600 documents from 4 classes. Tab. 9 shows that LDEBM performs the best when having only partial access to ground-truth data labels; it further validates the proposed formation for learning a well-structured latent space.

Table 9. Accuracy on AGNews. We report semi-supervised classification accuracy with varied number of labeled data.

MODEL	200	500	2500	10000
GLOVE-ID	70.4	78.0	84.1	87.1
GLOVE-OD	68.8	78.8	85.3	88.0
VAMPIRE	82.9	84.5	85.8	87.7
HARD EM	83.9	84.6	85.1	86.9
CATVAE	84.6	85.7	86.3	87.5
SVEBM	84.5	84.7	86.0	88.1
SVEBM-IB	<u>86.4</u>	<u>87.4</u>	<u>87.9</u>	<u>88.6</u>
<b>OURS</b>	<b>87.4</b>	<b>88.1</b>	<b>89.2</b>	<b>90.1</b>

## 5. Discussions and Related Work

**Text modeling** VAE has been one of the most prominent latent variable models for generative modeling (Kingma & Welling, 2013; Rezende et al., 2014). It is first applied to

text modeling in Bowman et al. (2016), followed by a wide range of work attacking challenging text generation problems using the shared framework of VAE. These include dialog generation (Serban et al., 2016; 2017; Wen et al., 2017; Zhao et al., 2017; 2018b; Fang et al., 2019), machine translation (Zhang et al., 2016), text summarization (Li et al., 2017a), and paraphrase generation (Gupta et al., 2018). In parallel, extensive efforts have been made to address issues like posterior collapse (Bowman et al., 2016; Higgins et al., 2016; Zhao et al., 2017; 2018a; He et al., 2018; Li et al., 2019; Fu et al., 2019) and mode-collapse (Shi et al., 2020) in training VAE to further improve the language modeling performance and text generation quality.

The interpretability of the generation process is naturally brought up as the generation quality achieves impressive progress. Recently, Zhao et al. (2018b), Shi et al. (2020), and Pang & Wu (2021) have explored interpretable text generation with deliberately designed latent spaces. Zhao et al. (2018b) use a discrete latent space to capture dialog actions; Shi et al. (2020) adopt a mixture of Gaussians as the VAE prior. To further improve the expressivity of latent space, Pang & Wu (2021) propose a symbol-vector coupling energy-based prior to learn a structured latent space. The coupling formulation provides a natural interface to induce the symbolic representation, which eliminates the need of training extra auxiliary inference networks for symbol induction. Our formulation inherits the advantages from Pang & Wu (2021) by choosing an appropriate symbol-vector coupling scheme and principally incorporating the IB. We further develop a geometric clustering-based regularization that complements the IB; it alleviates the mode-collapse problem in variational learning of the latent space model.

**Energy-based model** EBMs (Xie et al., 2016; Nijkamp et al., 2019; 2020; Han et al., 2020) have drawn growing interest in generative modeling. As an interesting branch, Pang et al. (2020a) learn an EBM in the latent space as a prior model for continuous latent variables; it greatly improves the expressivity over non-informative priors and demonstrates strong performance on downstream tasks, *e.g.*, image segmentation, molecule generation, and trajectory prediction (Yu et al., 2021; Pang et al., 2020b; 2021; Jing et al., 2019; 2018). However, both EBM and latent space EBM require MCMC sampling to learn the model. The degenerate sampling quality in practice can lead to poor generation quality and instability in training (Grathwohl et al., 2019; Du et al., 2021). We leverage diffusion models as a cure for the vanilla latent space EBM in this work; the proposed model shows reliable sampling quality in practice.

**Diffusion model** Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Gao et al., 2020), originating from Sohl-Dickstein et al. (2015), learn from a sequence of noise-

perturbed versions of the data. From such perturbed data, one can learn the conditional model to invert the diffusion process and generate high-quality samples given noisy inputs. On another front, Song & Ermon (2019; 2020); Song et al. (2020) extend the denoising score matching method (Vincent, 2011), modeling the diffusion process with continuous time step. Our formulation moves the model to the latent space in a variational framework with two benefits: (a) learning in a lower-dimensional space enables faster sampling and better convergence, and (b) learning the diffusion model in a continuous latent space avoids the discreteness of text data, which hinders the direct application of vanilla diffusion models to text modeling (Austin et al., 2021).

Similar to our work, Sinha et al. (2021), Nie et al. (2021), and Vahdat et al. (2021) have proposed to learn a diffusion model in the latent space. Specifically, Sinha et al. (2021) combine contrastive learning with diffusion models in the latent space of VAEs for controllable generation. Nie et al. (2021) and Vahdat et al. (2021) extend the idea of Song et al. (2020) in the latent space: Nie et al. (2021) perform controllable image generation by training a latent energy-based attribute classifier on a pre-trained generator; Vahdat et al. (2021) train score-based denoising diffusion models in the latent space of a powerful VAE (Vahdat & Kautz, 2020). Both methods have achieved very impressive image generation results. However, the listed methods are generally limited to image generation with tailored or pre-trained encoders and decoders. In contrast, our method is a general improvement of the sampling quality of latent space EBM; it is thus not restricted to a certain data modality. Moreover, our model equipped with the proposed techniques can be trained from scratch to form a well-structured latent space, in contrast to Vahdat et al. (2021) and Nie et al. (2021) which require a pre-learned latent space.

## 6. Conclusion and Future Works

We presented LDEBM, a novel symbiosis between symbol-vector coupling EBM and diffusion model that offers the best of both worlds. The proposed model shows reliable sampling quality, learns a well-structured and meaningful latent space from scratch, and can be flexibly extended to scenarios where data labels are available. It demonstrates superior performance over strong baselines on interpretable text modeling. We hope our work inspires future research along this challenging but promising research direction. A potential follow-up research problem is to reuse powerful pre-trained language models. One could consider integrating pre-trained models with our method to realize high-quality controllable generation at low computational cost.

**Acknowledgements:** Y. N. Wu was supported by NSF DMS-2015577. We would like to thank the anonymous reviewers for their constructive comments.

## References

- Ammanabrolu, P. and Riedl, M. O. Modeling worlds in text. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 6
- Austin, J., Johnson, D. D., Ho, J., Tarlow, D., and van den Berg, R. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 9
- Bengio, Y., Yao, L., Alain, G., and Vincent, P. Generalized denoising auto-encoders as generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2013. 3
- Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., and Bengio, S. Generating sentences from a continuous space. In *Conference on Computational Natural Language Learning (CoNLL)*, 2016. 1, 9
- Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., and Mercer, R. L. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 1993. 1
- Burda, Y., Grosse, R. B., and Salakhutdinov, R. Importance weighted autoencoders. In *International Conference on Learning Representations (ICLR)*, 2016. A3
- Card, D., Tan, C., and Smith, N. A. Neural models for documents with metadata. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018. 8
- Cho, K., van Merriënboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Annual Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014. A4
- Du, Y., Li, S., Tenenbaum, J., and Mordatch, I. Improved contrastive divergence training of energy based models. In *International Conference on Machine Learning (ICML)*, 2021. 9
- Eric, M., Krishnan, L., Charette, F., and Manning, C. D. Key-value retrieval networks for task-oriented dialogue. In *Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDial)*, 2017. 7
- Fang, L., Li, C., Gao, J., Dong, W., and Chen, C. Implicit deep latent variable models for text generation. In *Annual Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019. 1, 9
- Forgues, G., Pineau, J., Larchevêque, J.-M., and Tremblay, R. Bootstrapping dialog systems with word embeddings. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014. 7
- Fu, H., Li, C., Liu, X., Gao, J., Çelikyilmaz, A., and Carin, L. Cyclical annealing schedule: A simple approach to mitigating kl vanishing. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2019. 9
- Gao, R., Song, Y., Poole, B., Wu, Y. N., and Kingma, D. P. Learning energy-based models by diffusion recovery likelihood. In *International Conference on Learning Representations (ICLR)*, 2020. 2, 3, 5, 9
- Grathwohl, W., Wang, K.-C., Jacobsen, J.-H., Duvenaud, D., Norouzi, M., and Swersky, K. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations (ICLR)*, 2019. 2, 4, 9
- Gupta, A., Agarwal, A., Singh, P., and Rai, P. A deep generative framework for paraphrase generation. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018. 1, 9
- Gururangan, S., Dang, T., Card, D., and Smith, N. A. Variational pretraining for semi-supervised text classification. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019. A4
- Han, T., Nijkamp, E., Fang, X., Hill, M., Zhu, S.-C., and Wu, Y. N. Divergence triangle for joint training of generator model, energy-based model, and inferential model. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 4
- Han, T., Nijkamp, E., Zhou, L., Pang, B., Zhu, S.-C., and Wu, Y. N. Joint training of variational auto-encoder and latent energy-based model. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 9
- He, J., Spokoyny, D., Neubig, G., and Berg-Kirkpatrick, T. Lagging inference networks and posterior collapse in variational autoencoders. In *International Conference on Learning Representations (ICLR)*, 2018. 9
- Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations (ICLR)*, 2016. 9
- Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2, 4, 5, 9

- Jin, S., Wiseman, S., Stratos, K., and Livescu, K. Discrete latent variable representations for low-resource text classification. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020. [A4](#)
- Jing, M., Ma, X., Sun, F., and Liu, H. Learning and inferring movement with deep generative model. *arXiv preprint arXiv:1805.07252*, 2018. [9](#)
- Jing, M., Ma, X., Huang, W., Sun, F., and Liu, H. Task transfer by preference-based cost learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2019. [9](#)
- Johnson, J., Douze, M., and Jégou, H. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 2019. [5](#)
- Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. [A3](#)
- Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. [1](#), [3](#), [8](#), [A4](#)
- Kingma, D. P., Mohamed, S., Rezende, D. J., and Welling, M. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014. [A4](#)
- Li, B., He, J., Neubig, G., Berg-Kirkpatrick, T., and Yang, Y. A surprisingly effective fix for deep latent variable modeling of text. In *Annual Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019. [9](#)
- Li, J., Jia, R., He, H., and Liang, P. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2018. [7](#)
- Li, P., Lam, W., Bing, L., and Wang, Z. Deep recurrent generative decoder for abstractive text summarization. In *Annual Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017a. [1](#), [9](#)
- Li, Y., Su, H., Shen, X., Li, W., Cao, Z., and Niu, S. Daily-dialog: A manually labelled multi-turn dialogue dataset. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2017b. [7](#)
- Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. Building a large annotated corpus of english: the penn treebank. *Computational Linguistics*, 1993. [6](#)
- Merity, S., Keskar, N. S., and Socher, R. Regularizing and optimizing lstm language models. In *International Conference on Learning Representations (ICLR)*, 2018. [A3](#)
- Miao, Y., Yu, L., and Blunsom, P. Neural variational inference for text processing. In *International Conference on Machine Learning (ICML)*, 2016. [8](#)
- Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., and Khudanpur, S. Recurrent neural network based language model. In *Interspeech*, 2010. [6](#), [A4](#)
- Mitchell, J. and Lapata, M. Vector-based models of semantic composition. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2008. [7](#)
- Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2018. [A3](#)
- Nie, W., Vahdat, A., and Anandkumar, A. Controllable and compositional generation with latent-space energy-based models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. [2](#), [9](#)
- Nijkamp, E., Hill, M., Zhu, S.-C., and Wu, Y. N. Learning non-convergent non-persistent short-run mcmc toward energy-based model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. [2](#), [9](#)
- Nijkamp, E., Hill, M., Han, T., Zhu, S.-C., and Wu, Y. N. On the anatomy of mcmc-based maximum likelihood learning of energy-based models. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2020. [2](#), [9](#)
- Pang, B. and Wu, Y. N. Latent space energy-based model of symbol-vector coupling for text generation and classification. In *International Conference on Machine Learning (ICML)*, 2021. [1](#), [2](#), [3](#), [5](#), [6](#), [8](#), [9](#), [A3](#)
- Pang, B., Han, T., Nijkamp, E., Zhu, S.-C., and Wu, Y. N. Learning latent space energy-based prior model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020a. [2](#), [9](#), [A2](#)
- Pang, B., Han, T., and Wu, Y. N. Learning latent space energy-based prior model for molecule generation. *arXiv preprint arXiv:2010.09351*, 2020b. [9](#)
- Pang, B., Zhao, T., Xie, X., and Wu, Y. N. Trajectory prediction with latent belief energy-based model. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. [9](#)
- Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. Bleu: a method for automatic evaluation of machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002. [6](#)
- Pennington, J., Socher, R., and Manning, C. D. Glove: Global vectors for word representation. In *Annual Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014. [A3](#)

- Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning (ICML)*, 2014. 1, 8
- Rus, V. and Lintean, M. An optimal assessment of natural language student input using word-to-word similarity metrics. In *International Conference on Intelligent Tutoring Systems*, 2012. 7
- Serban, I., Sordoni, A., Bengio, Y., Courville, A., and Pineau, J. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2016. 1, 9
- Serban, I., Sordoni, A., Lowe, R., Charlin, L., Pineau, J., Courville, A., and Bengio, Y. A hierarchical latent variable encoder-decoder model for generating dialogues. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2017. 9
- Shi, W., Zhou, H., Miao, N., and Li, L. Dispersed exponential family mixture vaes for interpretable text generation. In *International Conference on Machine Learning (ICML)*, 2020. 1, 6, 7, 9, A3, A4
- Sinha, A., Song, J., Meng, C., and Ermon, S. D2c: Diffusion-denoising models for few-shot conditional generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 9
- Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning (ICML)*, 2015. 2, 3, 9
- Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 9
- Song, Y. and Ermon, S. Improved techniques for training score-based generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2, 9
- Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2020. 2, 9
- Subramanian, S., Rajeswar, S., Sordoni, A., Trischler, A., Courville, A., and Pal, C. Towards text generation with adversarially learned neural outlines. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. A4
- Tishby, N., Pereira, F. C., and Bialek, W. The information bottleneck method. *arXiv preprint physics/0004057*, 2000. 4
- Vahdat, A. and Kautz, J. NVAE: A deep hierarchical variational autoencoder. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 9
- Vahdat, A., Kreis, K., and Kautz, J. Score-based generative modeling in latent space. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 2, 9
- Vincent, P. A connection between score matching and denoising autoencoders. *Neural Computation*, 2011. 9
- Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.-A., and Bottou, L. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research (JMLR)*, 2010. A4
- Wang, W., Gan, Z., Xu, H., Zhang, R., Wang, G., Shen, D., Chen, C., and Carin, L. Topic-guided variational autoencoder for text generation. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2019. 1
- Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient langevin dynamics. In *International Conference on Machine Learning (ICML)*, 2011. 3, 5
- Wen, T.-H., Miao, Y., Blunsom, P., and Young, S. Latent intention dialogue models. In *International Conference on Machine Learning (ICML)*, 2017. 1, 9
- Xie, J., Lu, Y., Zhu, S.-C., and Wu, Y. A theory of generative convnet. In *International Conference on Machine Learning (ICML)*, 2016. 9
- Young, S., Gašić, M., Thomson, B., and Williams, J. D. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 2013. 1
- Yu, P., Xie, S., Ma, X., Zhu, Y., Wu, Y. N., and Zhu, S.-C. Unsupervised foreground extraction via deep region competition. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 9
- Zhang, B., Xiong, D., Su, J., Duan, H., and Zhang, M. Variational neural machine translation. In *Annual Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016. 9
- Zhang, X., Zhao, J., and LeCun, Y. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015. 8
- Zhao, J., Kim, Y., Zhang, K., Rush, A., and LeCun, Y. Adversarially regularized autoencoders. In *International Conference on Machine Learning (ICML)*, 2018a. 1, 6, 9

Zhao, T., Zhao, R., and Eskenazi, M. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2017. 1, 9

Zhao, T., Lee, K., and Eskenazi, M. Unsupervised discrete sentence representation learning for interpretable neural dialog generation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018b. 1, 6, 7, 9, A4

## A. Extended Derivations and Further Discussion

### A.1. Derivation of Conditional EBMs

We first define the marginal EBMs at each diffusion step:

$$\begin{cases} p_\alpha(\mathbf{z}_t) = \frac{1}{Z_{\alpha,t}} \exp(F_\alpha(\mathbf{z}_t, t)) p_0(\mathbf{z}_t), & t = T-1 \\ p_\alpha(\mathbf{z}_t) = \frac{1}{Z_{\alpha,t}} \exp(F_\alpha(\mathbf{z}_t, t)), & t = 0, 1, \dots, T-2 \end{cases} \quad (\text{A1})$$

where the marginal energy term is in a log-sum-exponential form  $F_\alpha(\mathbf{z}_t, t) = \log \sum_{\mathbf{y}} \exp(\langle \mathbf{y}, f_\alpha(\mathbf{z}_t, t) \rangle)$ ; it serves to aggregate the energy score from each category. Of note, the marginal EBM corresponding with the last diffusion step has a slightly different definition. We set this term as exponential tilting of a non-informative Gaussian prior  $p_0(\mathbf{z}_t)$  which helps to stabilize training in practice.

Recall that  $\mathbf{z}_{t+1} = \sqrt{1 - \sigma_{t+1}^2} \mathbf{z}_t + \sigma_{t+1} \epsilon_{t+1}$ . Let  $\tilde{\mathbf{z}}_t = \sqrt{1 - \sigma_{t+1}^2} \mathbf{z}_t$ . For  $t = 0, 1, \dots, T-2$ , we have

$$\begin{aligned} p_\alpha(\tilde{\mathbf{z}}_t | \mathbf{z}_{t+1}) &= \frac{p_\alpha(\tilde{\mathbf{z}}_t) p(\mathbf{z}_{t+1} | \tilde{\mathbf{z}}_t)}{p_\alpha(\mathbf{z}_{t+1})} \\ &= \frac{1}{\tilde{Z}_{\alpha,t}} \frac{\exp(F_\alpha(\tilde{\mathbf{z}}_t, t))}{p_\alpha(\mathbf{z}_{t+1})} \exp\left(-\frac{1}{2\sigma_{t+1}^2} \|\tilde{\mathbf{z}}_t - \mathbf{z}_{t+1}\|^2\right) \\ &= \frac{1}{\tilde{Z}_{\alpha,t}(\mathbf{z}_{t+1})} \exp\left(F_\alpha(\tilde{\mathbf{z}}_t, t) - \frac{1}{2\sigma_{t+1}^2} \|\tilde{\mathbf{z}}_t - \mathbf{z}_{t+1}\|^2\right), \end{aligned} \quad (\text{A2})$$

where  $\tilde{Z}_{\alpha,t} = (2\pi\sigma_{t+1}^2)^{\frac{n}{2}} Z_{\alpha,t}$ ; we slightly abuse the notation and use  $p(\mathbf{z}_{t+1} | \tilde{\mathbf{z}}_t)$  to represent the forward transition  $q(\mathbf{z}_{t+1} | \mathbf{z}_t)$  defined in Eq. (4) for notation consistency.

The diffused samples at time step  $T$  are close to Gaussian white noise;  $p_\alpha(\tilde{\mathbf{z}}_{T-1} | \mathbf{z}_T)$  therefore falls to its marginal distribution  $p(\tilde{\mathbf{z}}_{T-1})$  defined in Eq. (A1).

### A.2. Derivation of the ELBO

Recall that the ELBO in SVEBM is

$$\begin{aligned} \text{ELBO}_{\theta,\phi} &= \log p_\theta(\mathbf{x}) - \mathbb{D}_{\text{KL}}(q_\phi(\mathbf{z} | \mathbf{x}) \| p_\theta(\mathbf{z} | \mathbf{x})) \\ &= \mathbb{E}_{q_\phi(\mathbf{z} | \mathbf{x})} [\log p_\beta(\mathbf{x} | \mathbf{z})] - \mathbb{D}_{\text{KL}}(q_\phi(\mathbf{z} | \mathbf{x}) \| p_\alpha(\mathbf{z})) \\ &= \mathbb{E}_{q_\phi(\mathbf{z} | \mathbf{x})} [\log p_\beta(\mathbf{x} | \mathbf{z}) - \log q_\phi(\mathbf{z} | \mathbf{x}) + \log p_\alpha(\mathbf{z})], \end{aligned} \quad (\text{A3})$$

where  $\mathbb{D}_{\text{KL}}$  denotes the Kullback-Leibler divergence. Let us consider the full trajectory of the perturbed samples  $\mathbf{z}_0, \mathbf{z}_1, \dots, \mathbf{z}_T$ . The above equation can be written as

$$\begin{aligned} \text{ELBO}_{\theta,\phi} &= \mathbb{E}_{q_\phi(\mathbf{z}_0 | \mathbf{x})} [\log p_\beta(\mathbf{x} | \mathbf{z}_0) - \log q_\phi(\mathbf{z}_0 | \mathbf{x})] \\ &\quad + \mathbb{E}_{q_\phi(\mathbf{z}_0 | \mathbf{x})} \left[ \log \int_{\mathbf{z}_{1:T}} p_\alpha(\mathbf{z}_{0:T}) d\mathbf{z}_{1:T} \right], \end{aligned} \quad (\text{A4})$$

where the last term is further lower-bounded by introducing the forward trajectory distribution; the inequality holds by applying Jensen's Inequality:

$$\begin{aligned} &\mathbb{E}_{q_\phi(\mathbf{z}_0 | \mathbf{x})} \left[ \log \int_{\mathbf{z}_{1:T}} p_\alpha(\mathbf{z}_{0:T}) d\mathbf{z}_{1:T} \right] \\ &= \mathbb{E}_{q_\phi(\mathbf{z}_0 | \mathbf{x})} \left[ \log \int_{\mathbf{z}_{1:T}} q(\mathbf{z}_{1:T} | \mathbf{z}_0) \frac{p_\alpha(\mathbf{z}_{0:T})}{q(\mathbf{z}_{1:T} | \mathbf{z}_0)} d\mathbf{z}_{1:T} \right] \\ &\geq \mathbb{E}_{q_\phi(\mathbf{z}_0 | \mathbf{x})} \left[ \int_{\mathbf{z}_{1:T}} q(\mathbf{z}_{1:T} | \mathbf{z}_0) \log \frac{p_\alpha(\mathbf{z}_{0:T})}{q(\mathbf{z}_{1:T} | \mathbf{z}_0)} d\mathbf{z}_{1:T} \right] \\ &= \mathbb{E}_{q_\phi(\mathbf{z}_0 | \mathbf{x}), q(\mathbf{z}_{1:T} | \mathbf{z}_0)} \left[ \log \frac{p_\alpha(\mathbf{z}_{0:T})}{q(\mathbf{z}_{1:T} | \mathbf{z}_0)} \right]. \end{aligned} \quad (\text{A5})$$

Further, we can decompose the joint distribution of forward and backward trajectories as

$$\begin{aligned} &\mathbb{E}_{q_\phi(\mathbf{z}_0 | \mathbf{x}), q(\mathbf{z}_{1:T} | \mathbf{z}_0)} \left[ \log \frac{p_\alpha(\mathbf{z}_{0:T})}{q(\mathbf{z}_{1:T} | \mathbf{z}_0)} \right] = \\ &\mathbb{E}_{q_\phi(\mathbf{z}_0 | \mathbf{x}), q(\mathbf{z}_{1:T} | \mathbf{z}_0)} \left[ \log p(\mathbf{z}_T) + \sum_{t=0}^{T-1} \log \frac{p_\alpha(\mathbf{z}_t | \mathbf{z}_{t+1})}{q(\mathbf{z}_{t+1} | \mathbf{z}_t)} \right] = \\ &\mathbb{E} \left[ \log p(\mathbf{z}_T) + \sum_{t=0}^{T-1} \log p_\alpha(\mathbf{z}_t | \mathbf{z}_{t+1}) \right] + \sum_{t=0}^{T-1} \mathcal{H}(\mathbf{z}_{t+1} | \mathbf{z}_t), \end{aligned} \quad (\text{A6})$$

where  $p(\mathbf{z}_T)$  is standard Gaussian distribution;  $\mathbb{E}$  is the abbreviation of  $\mathbb{E}_{q_\phi(\mathbf{z}_0 | \mathbf{x}), q(\mathbf{z}_{1:T} | \mathbf{z}_0)}$ .  $\mathcal{H}(\mathbf{z}_{t+1} | \mathbf{z}_t)$ ,  $t = 0, \dots, 1$  is the conditional entropy under the forward trajectory distribution. We obtain  $\mathbf{z}_t$  by sampling  $\tilde{\mathbf{z}}_t$  from  $p_\alpha(\tilde{\mathbf{z}}_t | \mathbf{z}_{t+1})$  and then applying  $\mathbf{z}_t = \tilde{\mathbf{z}}_t / \sqrt{1 - \sigma_{t+1}^2}$ ; the reverse trajectory in our model is primarily defined by  $p_\alpha(\tilde{\mathbf{z}}_t | \mathbf{z}_{t+1})$  for  $t > 0$ . We use  $[\mathbf{z}_t | \mathbf{z}_{t+1}]$  to represent this process in the following sections; we may interchangeably use the notation of  $\tilde{\mathbf{z}}_t$  and  $\mathbf{z}_t$  for simplicity.

Note that the entropies can be analytically computed and do not involve learnable parameters. The joint training of inference, prior and generation models can be largely reduced to finding the agreement of the forward and reverse Markov transitions defined by  $q_\phi$  and  $p_\theta$  respectively.

### A.3. Detailed Discussion of Symbol Coupling

In Sec. 2, we briefly describe how to introduce the symbolic one-hot vector  $\mathbf{y}$ . Since only  $\mathbf{z}_0$  is connected with  $\mathbf{y}$ , we can first define the joint prior  $p_\alpha(\mathbf{y}, \mathbf{z}_0)$  as in Eq. (A1) by substituting  $F_\alpha(\tilde{\mathbf{z}}_0, 0)$  with  $\langle \mathbf{y}, f_\alpha(\tilde{\mathbf{z}}_0, 0) \rangle$ . Then the conditional symbol-vector coupling joint distribution follows as

$$\begin{aligned} p_\alpha(\mathbf{y}, \mathbf{z}_0 | \mathbf{z}_1) &= \frac{1}{\tilde{Z}_{\alpha,t=0}} \exp(\langle \mathbf{y}, f_\alpha(\tilde{\mathbf{z}}_0, 0) \rangle) \\ &\quad \exp\left(-\frac{1}{2\sigma_1^2} \|\tilde{\mathbf{z}}_0 - \mathbf{z}_1\|^2\right). \end{aligned} \quad (\text{A7})$$

Note that  $p_\alpha(\mathbf{y}, \mathbf{z}_0 | \mathbf{z}_1) = p_\alpha(\mathbf{y} | \mathbf{z}_0) p_\alpha(\mathbf{z}_0 | \mathbf{z}_1)$ , *i.e.*,  $\mathbf{z}_0$  is sufficient for inferring  $\mathbf{y}$  in this formulation:

$$\begin{aligned} p_\alpha(\mathbf{y} | \mathbf{z}_0, \mathbf{z}_1) &= \frac{p_\alpha(\mathbf{y}, \mathbf{z}_0 | \mathbf{z}_1)}{p_\alpha(\mathbf{z}_0 | \mathbf{z}_1)} \\ &= \frac{\exp(\langle \mathbf{y}, f_\alpha(\tilde{\mathbf{z}}_0, 0) \rangle)}{\exp(F_\alpha(\tilde{\mathbf{z}}_0, 0))}, \end{aligned} \quad (\text{A8})$$

so that given  $\mathbf{z}_0$ ,

$$p_\alpha(\mathbf{y} | \mathbf{z}_0) \propto \exp(\langle \mathbf{y}, f_\alpha(\tilde{\mathbf{z}}_0, 0) \rangle). \quad (\text{A9})$$

It similarly becomes a softmax classifier where  $f_\alpha(\tilde{\mathbf{z}}_0, 0)$  provides the logit scores for the  $K$  categories.

#### A.4. Derivation of the Information Bottleneck

We first define the mutual information term between  $\mathbf{z}_0$  and  $\mathbf{y}$ . Consider the joint distribution of  $\mathbf{x}, \mathbf{z}_0$  and  $\mathbf{y}$ ,  $\pi(\mathbf{y}, \mathbf{z}_0, \mathbf{x}) = p_\alpha(\mathbf{y} | \mathbf{z}_0) q_\phi(\mathbf{z}_0 | \mathbf{x}) q_{\text{data}}(\mathbf{x})$ ; the mutual information  $\mathcal{I}(\mathbf{z}_0, \mathbf{y})$  defined under  $\pi$  then follows as:

$$\begin{aligned} \mathcal{I}(\mathbf{z}_0, \mathbf{y}) &= \mathcal{H}(\mathbf{y}) - \mathcal{H}(\mathbf{y} | \mathbf{z}_0) \\ &= - \sum_{\mathbf{y}} q(\mathbf{y}) \log q(\mathbf{y}) \\ &\quad + \mathbb{E}_{q_\phi(\mathbf{z}_0)} \sum_{\mathbf{y}} p_\alpha(\mathbf{y} | \mathbf{z}_0) \log p_\alpha(\mathbf{y} | \mathbf{z}_0), \end{aligned} \quad (\text{A10})$$

where  $q(\mathbf{y}) = \mathbb{E}_{q_\phi(\mathbf{z}_0)} [p_\alpha(\mathbf{y} | \mathbf{z}_0)]$ ;  $p_\alpha(\mathbf{y} | \mathbf{z}_0)$  is the softmax probability over  $K$  categories in Eq. (A9).

We then show how to obtain the quantities defined in Sec. 3.2. For the marginal distribution of  $\mathbf{z}_0$ :

$$\begin{aligned} q_\phi(\mathbf{z}_0) &= \int_{\mathbf{x}, \mathbf{z}_{1:T}} Q_\phi(\mathbf{x}, \mathbf{z}_{0:T}) d\mathbf{x} d\mathbf{z}_{1:T} \\ &= \mathbb{E}_{q_{\text{data}}(\mathbf{x})} [q_\phi(\mathbf{z}_0 | \mathbf{x})]. \end{aligned} \quad (\text{A11})$$

The entropy and conditional entropy of  $\mathbf{z}_0$  are thus

$$\begin{aligned} \mathcal{H}(\mathbf{z}_0) &= -\mathbb{E}_{q_\phi(\mathbf{z}_0)} [\log q_\phi(\mathbf{z}_0)]; \\ \mathcal{H}(\mathbf{z}_0 | \mathbf{x}) &= -\mathbb{E}_{Q_\phi(\mathbf{x}, \mathbf{z}_0)} [\log q_\phi(\mathbf{z}_0 | \mathbf{x})]. \end{aligned} \quad (\text{A12})$$

Taking together, we can then decompose the KL-Divergence,  $\mathbb{D}_{\text{KL}}(Q_\phi \| P_\theta)$ , in Eq. (8) as:

$$\begin{aligned} \mathbb{D}_{\text{KL}}(Q_\phi \| P_\theta) &= \mathbb{E}_{Q_\phi} [q_{\text{data}}(\mathbf{x})] + \mathbb{E}_{Q_\phi} [q_\phi(\mathbf{z}_{0:T} | \mathbf{x})] \\ &\quad - \mathbb{E}_{Q_\phi} [p_\alpha(\mathbf{z}_{0:T})] - \mathbb{E}_{Q_\phi} [p_\beta(\mathbf{x} | \mathbf{z}_0)], \end{aligned} \quad (\text{A13})$$

and further as:

$$\begin{aligned} -\mathcal{H}(\mathbf{x}) &+ \sum_{t=0}^{T-1} \mathcal{H}(\mathbf{z}_{t+1} | \mathbf{z}_t) - \mathcal{H}(\mathbf{z}_0 | \mathbf{x}) + \mathcal{H}(\mathbf{z}_0) - \mathcal{H}(\mathbf{z}_0) \\ &- \mathbb{E}_{Q_\phi} [p_\alpha(\mathbf{z}_{0:T})] - \mathbb{E}_{Q_\phi} [p_\beta(\mathbf{x} | \mathbf{z}_0)], \end{aligned} \quad (\text{A14})$$

by plugging in  $\mathcal{H}(\mathbf{z}_0) - \mathcal{H}(\mathbf{z}_0) = 0$ . Rearranging Eq. (A14), we can obtain

$$\begin{aligned} \mathbb{D}_{\text{KL}}(Q_\phi \| P_\theta) &= \mathcal{C} - \mathbb{E}_{Q_\phi} [p_\beta(\mathbf{x} | \mathbf{z}_0)] \\ &\quad + \mathbb{D}_{\text{KL}}(q_\phi(\mathbf{z}_0) \| p_\alpha(\mathbf{z}_{0:T})) + \mathcal{I}(\mathbf{x}, \mathbf{z}_0), \end{aligned} \quad (\text{A15})$$

which leads to our result in Eq. (9).

#### A.5. Derivation of the Learning Gradient

Recall that we derive the extended version of Eq. (6) in Appx. A.2. To calculate the gradient of  $\alpha$ , we have

$$\begin{aligned} \nabla_\alpha \text{ELBO}_{\text{Diff}, \theta, \phi} &= \nabla_\alpha \mathbb{E} \left[ \sum_{t=0}^{T-1} \log p_\alpha(\mathbf{z}_t | \mathbf{z}_{t+1}) \right] \\ &= \mathbb{E} \left[ \sum_{t=0}^{T-1} \nabla_\alpha \log p_\alpha(\mathbf{z}_t | \mathbf{z}_{t+1}) \right], \end{aligned} \quad (\text{A16})$$

where  $\mathbb{E}$  is the abbreviation of  $\mathbb{E}_{q_\phi(\mathbf{z}_0 | \mathbf{x}), q(\mathbf{z}_{1:T} | \mathbf{z}_0)}$ ; in practice, we use Monte-Carlo average to approximate the expectation. We next examine the learning gradient for each diffusion step  $t$ .

$$\nabla_\alpha \log p_\alpha(\mathbf{z}_t | \mathbf{z}_{t+1}) = \nabla_\alpha F_\alpha(\tilde{\mathbf{z}}_t, t) - \nabla_\alpha \tilde{Z}_{\alpha, t}(\mathbf{z}_{t+1}), \quad (\text{A17})$$

where the quadratic term  $\frac{1}{2\sigma_{t+1}^2} \|\tilde{\mathbf{z}}_t - \mathbf{z}_{t+1}\|^2$  is not related to  $\alpha$  and gets cancelled. According to the definition of the partition function in Sec. 2, we can similarly derive

$$\nabla_\alpha \tilde{Z}_{\alpha, t}(\mathbf{z}_{t+1}) = \mathbb{E}_{p_\alpha(\tilde{\mathbf{z}}_t | \mathbf{z}_{t+1})} [\nabla_\alpha F_\alpha(\tilde{\mathbf{z}}_t, t)], \quad (\text{A18})$$

as in Pang et al. (2020a). For the prior model, we thus have

$$\begin{aligned} \nabla_\alpha \text{ELBO}_t &= \mathbb{E}_{q_\phi(\tilde{\mathbf{z}}_t, \mathbf{z}_0 | \mathbf{x})} [\nabla_\alpha F_\alpha(\tilde{\mathbf{z}}_t, t)] \\ &\quad - \mathbb{E}_{q_\phi(\mathbf{z}_{t+1}, \mathbf{z}_0 | \mathbf{x}), p_\alpha(\tilde{\mathbf{z}}_t | \mathbf{z}_{t+1})} [\nabla_\alpha F_\alpha(\tilde{\mathbf{z}}_t, t)], \end{aligned} \quad (\text{A19})$$

where  $q_\phi(\tilde{\mathbf{z}}_t, \mathbf{z}_0 | \mathbf{x}) = q(\tilde{\mathbf{z}}_t | \mathbf{z}_0) q_\phi(\mathbf{z}_0 | \mathbf{x})$ . Note that we can sample  $\mathbf{z}_t$ ,  $t > 0$  directly from

$$q(\mathbf{z}_t | \mathbf{z}_0) = \mathcal{N}(\mathbf{z}_t; \sqrt{\gamma_t} \mathbf{z}_{t-1}, (1 - \gamma_t) \mathbf{I}), \quad (\text{A20})$$

by merging the Gaussian noises during forward diffusion process; we denote  $\gamma_t = 1 - \sigma_t^2$  and  $\bar{\gamma}_t = \prod_{i=1}^t \gamma_i$ .

For the encoder and decoder, based on Eq. (6) and Eq. (A6), we have

$$\begin{aligned} \nabla_\psi \text{ELBO} &= \nabla_\psi \mathbb{E}_{q_\phi(\mathbf{z}_0 | \mathbf{x})} [\log p_\beta(\mathbf{x} | \mathbf{z}_0) - \log q_\phi(\mathbf{z}_0 | \mathbf{x})] \\ &\quad - \nabla_\phi \mathbb{E}_{q_\phi(\mathbf{z}_{0:T} | \mathbf{x})} \left[ \log p(\mathbf{z}_T) + \sum_{t=0}^{T-1} \log p_\alpha(\mathbf{z}_t | \mathbf{z}_{t+1}) \right], \end{aligned} \quad (\text{A21})$$

where the summation of energy terms provides extra guidance for the optimization of encoder.

Table A1. Network architecture for the LDEBM prior.  $N$  is set to 12 for all the experiments.

Layers	Output size	Note
<b>Time Embedding</b>		
Input: $t$	1	Index of diffusion step
Sin. embedding	200	
Linear, LReLU	200	negative_slope 0.2
Linear	200	
<b>Input Embedding</b>		
Input: $\mathbf{z}$	$d_{\text{lat}}$	
Linear, LReLU	200	negative_slope 0.2
Linear	200	
<b>Context Embedding</b> (for response generation only)		
Input: $\mathbf{z}_{\text{ctx}}$	512	ctx. embedding
Linear, LReLU	200	negative_slope 0.2
Linear	200	
<b>LDEBM Prior</b>		
Input: $\mathbf{z}, t$	1, $d_{\text{lat}}$	optional $\mathbf{z}_{\text{ctx}}$
* $\mathbf{z}_{\text{ctx}}$	512	
Embedding	200	Embedding of each input
Concatenate	400	w/o ctx.
	600	w/ ctx.
LReLU, Linear	200	negative_slope 0.2
N ResBlocks	200	LReLU, Linear + Input
LReLU, Linear	$K$	$K$ class logits
Log-Sum-Exp	1	energy score

## B. Extra Experiment Details and Discussion

### B.1. Network Architecture and Hyperparameters

We provide detailed network architecture for the latent space model of this work in Tab. A1; we adopt the same architecture throughout the experiments. Spectral normalization (Miyato et al., 2018) is used to regularize parameters in linear layers. The encoder and decoder in all models are the same as in Pang & Wu (2021), implemented with a single-layer GRU with a hidden size of 512. The key hyperparameters of LDEBM for each dataset are listed in Tab. A2. Of note, we use the same dimension of the latent space as in (Pang & Wu, 2021) for a fair comparison.

$\lambda_1$  is the hyperparameter that reweights the term in Eq. (A6); it generally controls how fast  $q_\phi$  and  $p_\theta$  run towards each other.  $\lambda_2$  refers to the hyperparameter in Eq. (9); it controls the trade-off between the compressivity of  $\mathbf{z}_0$  about  $\mathbf{x}$  and its expressivity to  $\mathbf{y}$ .  $\lambda_3$  controls the weight of classification loss mentioned in Sec. 3.3; recall that we use pseudo-label  $\hat{\mathbf{y}}$  inferred by the geometric clustering algorithm or the ground-truth label  $\mathbf{y}$  to supervise  $p_\alpha(\mathbf{y}|\mathbf{z}_0)$  in our modeling. For controllable generation and semi-supervised classification,

Table A2. Hyperparameters of LDEBM. DD-CLS presents the set of hyperparameters used in unsupervised clustering on DD dataset. DD-GEN presents the set of hyperparameters used in conditional response generation on DD dataset.

DATASET	$d_{\text{lat}}$	$K$	$\lambda_1$	$\lambda_2$	$\lambda_3$
2D GAUSSIAN	2	16	1	0.05	0.05
2D PINWHEEL	2	10	1	0.05	0.05
PTB	40	20	0.1	0.05	0.05
JERICHO	40	20	0.1	0.05	0.05
DD-CLS	32	125	0.01	0.05	0.5
DD-GEN	32	125	1	0.05	0.05
SMD	32	125	10	10	5
YELP	40	2	50	50	200
AGNEWS	20	4	1e-3	5	200

we find it important to have a larger weight on the classification loss so that the model is forced to capture the major modes of the data.

For optimization, we use Adam optimizer (Kingma & Ba, 2014) with  $\beta_1 = 0.9$  and  $\beta_2 = 0.999$  for all the experiments. On all the datasets but 2D synthetic datasets and AGNews dataset, we use a batch size of 128 and a constant learning rate of  $1e-3$  for encoder and decoder without weight decay. For LDEBM, we use a constant learning rate of  $1e-4$ . We use a larger batch size of 1000 on 2D synthetic datasets. On the AGNews dataset, we use the same set of hyperparameters as in Pang & Wu (2021) for optimization. The batch size is set to 200; the initial learning rate is  $1e-4$  for encoder and decoder, and  $1e-5$  for LDEBM. Learning rates are exponentially decayed with a decay rate of 0.998 for each model. Encoder and LDEBM have a weight decay rate of  $2e-3$  and  $1e-3$ , respectively.

### B.2. Experiment Settings and Baselines

**Experiment settings** For generative modeling, following previous methods (Shi et al., 2020; Pang & Wu, 2021), the NLL term is computed with importance sampling (Burda et al., 2016) using 500 importance samples. To compute rPPL, we set the generated sample size as 40,000, which is the same size as PTB training set. We recruit ASGD Weight-Dropped LSTM (Merity et al., 2018) to compute rPPL as in previous works.

In terms of conditional response generation, for word-embedding-based evaluation on SMD and DD, we use the publicly available GloVe (Pennington et al., 2014) word embeddings of 300 dimension trained on 840B tokens, and report the score from 1 response per context. We use a context window size of 5 during training and evaluation.

The maximum length of each sentence is set to 40 words for most datasets and 70 words for the JerichoWorld dataset. On JerichoWorld dataset, we extract the description of each state as the text data.

**Baselines** On **PTB**, **DD** and **SMD**, our model is compared with the following baselines: (1) RNNLM (Mikolov et al., 2010), the language model implemented with GRU (Cho et al., 2014); (2) AE (Vincent et al., 2010), the deterministic auto-encoder which has no regularization to the latent space; (3) DAE, the AE with a discrete latent space; (4) VAE (Kingma & Welling, 2013), the vanilla VAE with a continuous latent space and a non-informative Gaussian prior; (5) DVAE, the VAE with a discrete latent space; (6) DI-VAE (Zhao et al., 2018b), a DVAE variant with a mutual information term between the observed piece of text  $x$  and its inferred latent variable  $z$ ; (7) semi-VAE (Kingma et al., 2014), the semi-supervised VAE model with independent discrete and continuous latent variables; (8) GM-VAE, the VAE with a Gaussian mixture prior; (9) DGM-VAE (Shi et al., 2020), the GM-VAE with a dispersion term that avoids the mode-collapse of Gaussian mixture prior; (10) semi-VAE +  $\mathcal{I}(x, y)$ , GM-VAE +  $\mathcal{I}(x, y)$ , DGM-VAE +  $\mathcal{I}(x, y)$ , are the same models as (7), (8), and (9) respectively, but with a mutual information term between  $x$  and  $y$  computed using separate inference networks for  $y$  and  $z$ . We compare with the close competitors (11) SVEBM, the symbol-vector coupling prior model and (12) SVEBM-IB, SVEBM with regularization based on information-bottleneck.

On **Yelp** dataset, we additionally include text conditional GAN (Subramanian et al., 2018) as a baseline for controllable generation. On **AGNews** dataset, we further compare our model to VAMPIRE (Gururangan et al., 2019), a VAE-based semi-supervised text learning model. Other baselines include its supervised learning variants: (1) the model trained with Glove embedding pre-trained on 840 billion words (Glove-OD); (2) the model trained with Glove embedding on in-domain unlabeled data (Glove-ID). We also include more recent baselines such as Hard EM and CatVAE (Jin et al., 2020) that improve over VAMPIRE.

### B.3. Extra Details for Experiments

**More ablation study** We conduct additional experiments on both PTB and DD datasets to inspect the contribution of the proposed techniques. In Sec. 4.1, we have reported results on PTB and datasets of OURS w/o GC which represents the model with Information Bottleneck but without Geometric Clustering (GC); OURS denotes the full model.

We further conduct experiments on the proposed model without using IB or GC. We observe that the proposed model using only diffusion-based sampling scheme has a rPPL of 166.26, BLEU of 11.30, wKL of 0.07 and NLL of 80.76 on PTB; it has a MI of 0.01, BLEU of 19.28, Act. of 0.12 and Emo. of 0.06 on DD, which is better than SVEBMs (please see Tabs. 1 and 3 in Sec. 4.1).

We also add GC to SVEBM (denoted as SVE-IB w/ GC). We find that SVE-IB w/ GC does perform better compared

with SVE-IB, showing a rPPL of 179.95, BLEU of 10.08, wKL of 0.15 and NLL of 93.28 on PTB; it has a MI of 2.88, BLEU of 11.75, Act. of 0.61 and Emo. of 0.60 on DD. Notably, SVE-IB w/ GC is still inferior to LDEBMs.

In summary, we think these additional experiments (1) emphasize the importance of our diffusion-based modeling approach, and (2) demonstrate the effectiveness of GC as additional regularization.

**2D synthetic data** We provide the full evolution of SVEBM-IB and our models as visualized in Fig. A2. Though SVEBM-IB can capture some regularities of the data in the early stages of training, the model is prone to collapse due to the degenerated sampling quality. This features an exploding KL-term and leads to poor performance on generation. Our preliminary experiments indicate that common deep learning heuristics for improving the model capacity barely help. These include but are not limited to increasing the number of parameters in SVEBM, *i.e.*, using larger models, and adopting deliberately designed activation functions or normalization modules. LDEBM w/o geometric clustering has a better sampling quality and effectively mitigates the instability in training. However, the mode coverage is not satisfying in data space; the structure is unclear in latent space. In contrast, LDEBM w/ geometric clustering shows superior generation quality with better mode coverage. It demonstrates a better-structured latent space.

**Sentence completion** To perform sentence completion, we adopt a two-stage training scheme. We first train the LDEBM with inference, prior and generation models on the JerichoWorld dataset. After the first-stage training, the parameters of prior, inference and generation models are fixed. We then train a shallow MLP in the latent space to project the inferred posterior  $z_0$  to a disentangled space; the variables in the projected  $z_0$  can be grouped as: (a) the representation of observable words  $\hat{z}_{\text{obs}}$  in the input sentence and (b) the representation of unknown words  $\hat{z}_{\text{unk}}$ . Conditional sampling in the latent space then refers to updating  $\hat{z}_{\text{unk}}$  based on the fixed  $\hat{z}_{\text{obs}}$  by running Langevin dynamics guided by the latent space model.

We mask half of the words in the sentences with `<unk>` token to prepare the inputs. In the second stage of training, we supervise the MLP by minimizing the reconstruction error between only the observable words of the input the sentence and the corresponding outputs of the model.

**Sentence sentiment control** Recall that in our formulation only  $z_0$  is connected to  $y$ . We therefore condition only the final reverse diffusion step  $[z_0|z_1]$  on  $y$  when performing controllable generation, *i.e.*, using  $y$  to guide the generation only when  $t = 0$  in Alg. 2. This can be a bit counter-intuitive since no label information is injected in previous reverse

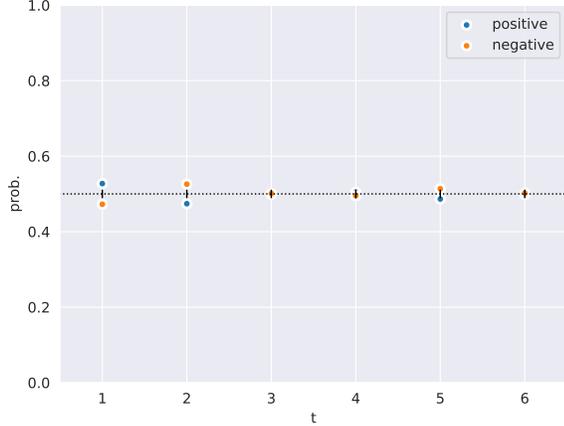


Figure A1. **Visualization of  $p_\alpha(\mathbf{y}|\mathbf{z}_t)$  over  $t$ .**  $p_\alpha(\mathbf{y}|\mathbf{z}_t)$  is constantly around the probability of 0.5 over  $t$ .

steps. Theoretically,  $\mathbf{y}$  and  $\mathbf{z}_{1:T}$  are independent given  $\mathbf{z}_0$  in our formulation; however, we empirically observe that  $\mathbf{y}$  and  $\mathbf{z}_t$  for  $t > 0$  are nearly independent even marginally after we integrating out  $\mathbf{z}_{0:t-1}$  in our model. In other words,  $p_\alpha(\mathbf{y}|\mathbf{z}_t)$ ,  $t > 0$  are in general non-informative since adding noise in the latent space could be much more corrupting than adding noise in the data space. The model learns to enjoy the less multi-modal energy landscape in previous reverse steps; it then seeks the given mode only in the most informative final reverse step. We examine  $p_\alpha(\mathbf{y}|\mathbf{z}_t)$ ,  $t > 0$  for the model trained on Yelp dataset by marginalizing out  $\mathbf{z}_{t-1}$  of  $p_\alpha(\mathbf{y}, \mathbf{z}_{t-1}|\mathbf{z}_t)$ ,  $t > 0$ . For example, for  $t = 1$ , we may calculate

$$\begin{aligned}
 p_\alpha(\mathbf{y}|\mathbf{z}_1) &= \int_{\mathbf{z}_0} p_\alpha(\mathbf{y}|\mathbf{z}_0)p_\alpha(\mathbf{z}_0|\mathbf{z}_1)d\mathbf{z}_0 \\
 &= \mathbb{E}_{p_\alpha(\mathbf{z}_0|\mathbf{z}_1)} [p_\alpha(\mathbf{y}|\mathbf{z}_0)] \\
 &\approx \frac{1}{M} \sum_{i=1}^M p_\alpha(\mathbf{y}|\mathbf{z}_0^{(i)}).
 \end{aligned} \tag{A22}$$

See Fig. A1 for the visualization of  $p_\alpha(\mathbf{y}|\mathbf{z}_t)$  over  $t$ .

A more intuitive method is to use the data label  $\mathbf{y}$  to supervise each  $[\mathbf{y}, \mathbf{z}_t|\mathbf{z}_{t+1}]$ , so that we can propagate the label information through the whole trajectory. Given  $\mathbf{z}_0$ ,  $\mathbf{y}$  and  $\mathbf{z}_{1:T}$  are independent. But if we marginalize out  $\mathbf{z}_0$ ,  $\mathbf{y}$  will depend on  $\mathbf{z}_1$ . Similarly, if we continue to marginalize out  $\mathbf{z}_1$ ,  $\mathbf{y}$  will depend on  $\mathbf{z}_2$ . Repeating this process results in  $p_\alpha(\mathbf{y}|\mathbf{z}_t)$  for each  $t$  after integrating out  $\mathbf{z}_{0:t-1}$ . Supervising  $p_\alpha(\mathbf{y}|\mathbf{z}_t)$ ,  $t > 0$  using  $\mathbf{y}$  therefore effectively encodes the label information into the whole trajectory.

While the marginalization can be difficult, we may approximate it by learning the amortized version of  $p_\alpha(\mathbf{y}|\mathbf{z}_t)$ ,  $t > 0$  as  $p_\alpha(\mathbf{y}, \mathbf{z}_{t-1} = \mu_{\phi,t-1}|\mathbf{z}_t)$ ,  $t > 0$ , where  $\mu_{\phi,t}$  is the posterior mean of  $\mathbf{z}_t$ . We may therefore circumvent the intractable integration in practice and guide the whole trajectory for controllable generation.

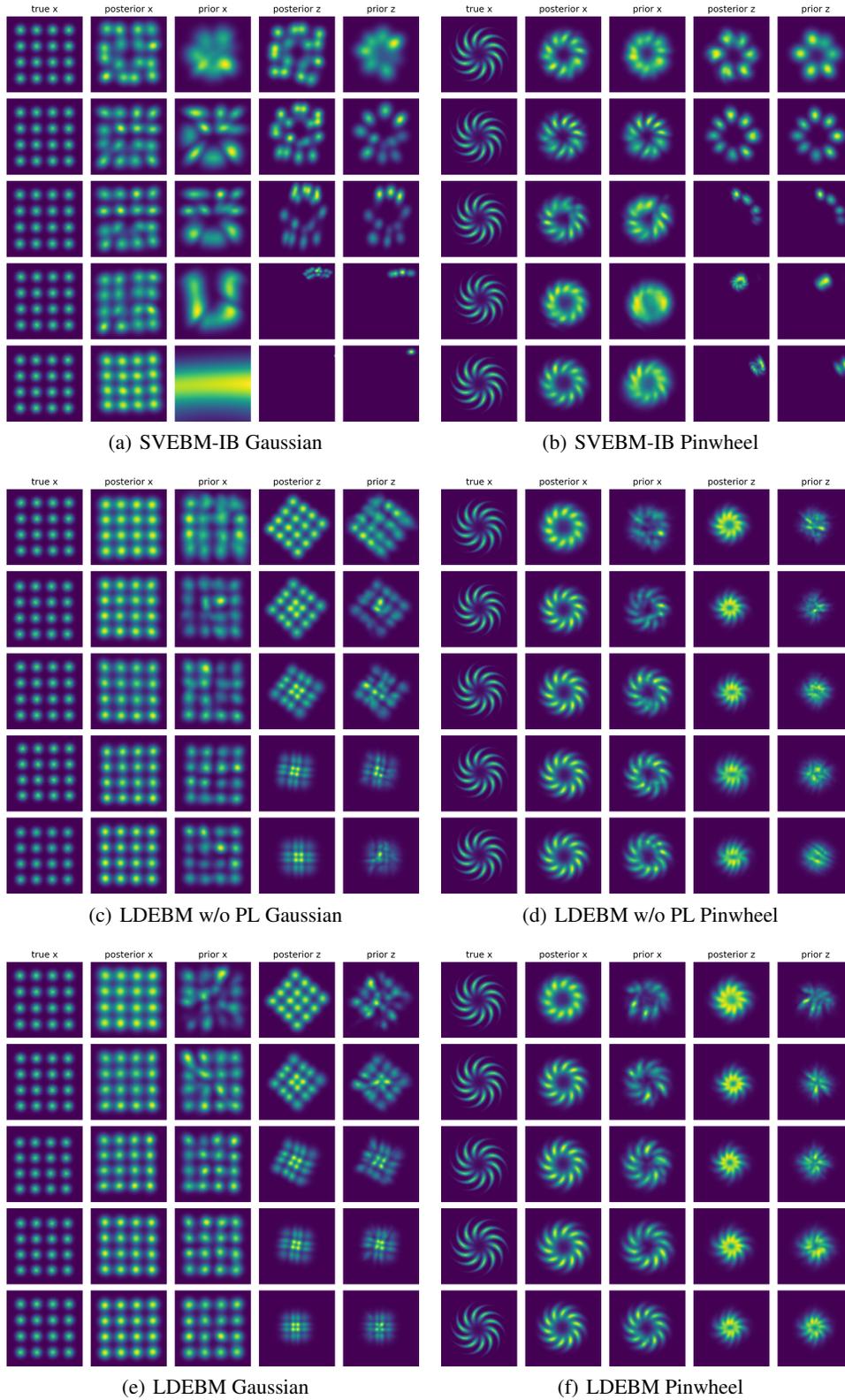


Figure A2. Full evolution of SVEBM-IB and our models. In each sub-figure, we provide the typical states of the model trained on the corresponding dataset, sequentially from the top row to the bottom row.