Generative Text Modeling through Short Run Inference

Bo Pang* UCLA bopang@ucla.edu
Erik Nijkamp* UCLA enijkamp@ucla.edu
Tian Han Stevens Institute of Technology than6@stevens.edu
Ying Nian Wu UCLA ywu@stat.ucla.edu

Abstract

Latent variable models for text, when trained successfully, accurately model the data distribution and capture global semantic and syntactic features of sentences. The prominent approach to train such models is variational autoencoders (VAE). It is nevertheless challenging to train and often results in a trivial local optimum where the latent variable is ignored and its posterior collapses into the prior, an issue known as *posterior collapse*. Various techniques have been proposed to mitigate this issue. Most of them focus on improving the inference model to yield latent codes of higher quality. The present work proposes a short run dynamics for inference. It is initialized from the prior distribution of the latent variable and then runs a small number (e.g., 20) of Langevin dynamics steps guided by its posterior distribution. The major advantage of our method is that it does not require a separate inference model or assume simple geometry of the posterior distribution, thus rendering an automatic, natural and flexible inference engine. We show that the models trained with short run dynamics more accurately model the data, compared to strong language model and VAE baselines, and exhibit no sign of posterior collapse. Analyses of the latent space show that interpolation in the latent space is able to generate coherent sentences with smooth transition and demonstrate improved classification over strong baselines with latent features from unsupervised pretraining. These results together expose a well-structured latent space of our generative model.

1 Introduction

The state-of-the-art language models (LM) are often modeled with recurrent neural networks (RNN) (Mikolov et al., 2010) or attention-based models (Dong et al., 2019; Vaswani et al., 2017). They are optimized by making a series of next-step predictions, encouraging the models to capture local dependency rather than global semantic features or high-level syntactic properties. A seminal work by Bowman et al. (2016) extends the standard LM to incorporate a continuous latent space which is aimed to explicitly capture global features. They formulate and train the model as a varational autoencoder (VAE) (Kingma and Welling, 2014). Indeed, the model is able to generate coherent and diverse sentences through continuous sampling, and provide smooth interpolation between sentences, uncovering a well-formed latent space.

However, training VAE for text is challenging and often leads to a trivial local optimum, *posterior collapse*. Specifically, the training objective of VAE can be decomposed into a reconstruction term and a KL term that regularizes the distance between the posterior and prior of the latent variable. Due to the autoregressive nature of the decoder, it is able to reconstruct the data well by simply relying on the one-step-ahead groud-truth and evolving model state while completely ignoring the latent codes. The posterior hence collapses into the prior, carrying no information. This is an important open question in this field. As pointed out in Fu et al. (2019), two paths work together to generate sentences in VAE. One path (Path A) is through the latent codes, while the other (Path B) is conditioned on the prior ground-truth or previously generated tokens. The posterior collapse describes an easy solution, that is, relying on Path B and ignoring Path A. Prior efforts made to address this issue by and large are along the two paths. One can control the information available from Path B to force the decoder to employ more Path A information. Bowman et al. (2016) dropout the input words to the decoder and Yang et al. (2017) utilize a dilated CNN to con-

*Equal contributions.

trol the size of context from previously generated words. Along Path A, various techniques have been developed to improve the latent code quality. Bowman et al. (2016) anneal the weight of the KL term from a small number to reduce the regularization in the beginning of the training (Anneal-VAE), while Fu et al. (2019) further propose to adopt a cyclical annealing schedule (Cyclical-VAE). He et al. (2019) update the encoder multiple times before one decoder update (Lagging-VAE). Li et al. (2019) initialize the VAE with an autoencoder (AE) and adopt a hinge loss for the KL term such that KL is not driven down below a target rate (FBP-VAE and FB-VAE). These techniques fall under the framework of amortized variational inference. Despite its fast inference, Cremer et al. (2018) observes that an amortization gap, the gap between the log-likelihood and the ELBO, can be large. Thus Kim et al. (2018) proposes semi-amortized variational autoencoders (SA-VAE) in which initial variational parameters are obtained from an encoder as in VAE, and the ELBO is then optimized with respect to the variational parameters to refine them.

An alternative to variational inference is Markov chain Monte Carlo (MCMC) sampling. MCMC posterior sampling may be in the form of Langevin dynamics (Langevin, 1908) or Hamiltonian Monte Carlo (HMC) (Neal, 2011; Chen et al., 2014). Traditional MCMC can be time-consuming as the Markov chains require a long running time, each iteration involving a gradient computation through the decoder.

In this article, we propose to apply a short run inference (SRI) dynamics, such as finite step Langevin dynamics, guided by the posterior distribution of the latent variable as an approximate inference engine. For each training example, we initialize such a short run dynamics from the prior distribution such as Gaussian noise distribution, and run a finite number (e.g., 20) of steps of updates. This amounts to a residual network which transforms the initial noise distribution to an approximate posterior distribution.

One major advantage of the SRI is that it is natural and automatic. Designing and tuning a separate inference model is not a trivial task. In prior work, the inference model requires careful tuning to avoid posterior collapse in VAEs for text modeling. For instance, the inference model needs to be aggressively trained (He et al., 2019), pre-trained with an autoencoder (Li et al., 2019), or refined with gradient descent guided by the ELBO (Kim et al., 2018). In contrast, the short run dynamics guided by the log-posterior of the latent variable can be automatically obtained on modern deep learning platforms. In addition, our method does not assume a closed-form density for the posterior, like a Gaussian with diagonal covariance matrix, and hence are possible to have a good approximate posterior and provide good latent code. Lastly, we optimize the hyper-parameter of the short run dynamics by minimizing the KL divergence between the short-run-dynamics-induced posterior and the true posterior, to further improve the approximate posterior.

Empirically, we show that the model trained with the SRI is able to outperform a standard LSTM language model by employing an LSTM generative model, while exhibiting active utilization of the latent space, improving over models trained with VAE-based approaches. Moreover, we find the learned latent space is smooth, allowing for coherent and smooth interpolation and reconstruction from noisy samples, and captures sufficient global information, enabling enhanced classification accuracy over state-of-the-art baselines.

In summary, the following are contributions of our paper. (1) We propose to use short run inference dynamics to train generative models for sentences without the need for an auxiliary inference network. (2) We demonstrate that the generative model trained with the SRI is able to accurately model the data distribution and make active use of the latent space, exhibiting no sign of posterior collapse. (3) We show that the learned latent space is smooth and captures rich global representations of sentences.

2 Model and learning algorithm

2.1 Generative model

Let x be the observed example, such as a sentence. Let z be the latent variable. We may consider z as forming an interpretation or explanation of x, such as the global semantics and/or high-level syntactic properties of sentences. Consider the following generative model for x,

$$z \sim p(z) \quad x \sim p_\theta(x|z). \tag{1}$$

where $p(z)$ is the prior and $p_\theta(x|z)$ is given by a generative model parameterized with θ. The marginal distribution of x is $p_\theta(x) = \int p_\theta(x, z)dz$. Given x, the inference of z can be based on the posterior distribution $p_\theta(z|x) = p_\theta(x, z)/p_\theta(x)$.

2.2 Learning and inference

Let $p_{\text{data}}(x)$ be the data distribution that generates the example x. The learning of parameters θ of $p_\theta(x)$ can be based on $\min_\theta \text{KL}(p_{\text{data}}(x)\|p_\theta(x))$, where $\text{KL}(p\|q) = \text{E}_p[\log(p(x)/q(x))]$ is the Kullback-Leibler divergence between p and q (or from p to q since $\text{KL}(p\|q)$ is asymmetric). If we observe training examples $\{x_i, i = 1, ..., n\} \sim p_{\text{data}}(x)$, the above minimization can be approximated by maximizing the log-likelihood

$$L(\theta) = \frac{1}{n}\sum_{i=1}^{n} \log p_\theta(x_i), \tag{2}$$

which leads to the maximum likelihood estimate (MLE).

The gradient of the log-likelihood, $L'(\theta)$, can be computed according to the following identity:

$$\begin{aligned}
\frac{\partial}{\partial\theta}\log p_\theta(x) &= \frac{1}{p_\theta(x)}\frac{\partial}{\partial\theta}p_\theta(x) \\
&= \frac{1}{p_\theta(x)}\int \frac{\partial}{\partial\theta}p_\theta(x, z)dz \\
&= \int \frac{\partial}{\partial\theta}\log p_\theta(x, z)\frac{p_\theta(x, z)}{p_\theta(x)}dz \\
&= \text{E}_{p_\theta(z|x)}\left[\frac{\partial}{\partial\theta}\log p_\theta(x, z)\right]. \quad (3)
\end{aligned}$$

While the marginal distribution $p(x) = \int p(x|z)p(z)dz$ is intractable due to the latent variables z being integrated out, the above expectation can be approximated by Monte Carlo average with samples drawn from $p_\theta(z|x)$. Such samples from $p_\theta(z|x)$ can be obtained by MCMC in the form of Langevin dynamics (Langevin, 1908), which iterates

$$z_{k+1} = z_k + s\frac{\partial}{\partial z}\log p_\theta(z_k|x) + \sqrt{2s}\epsilon_k, \tag{4}$$

where $\epsilon_k \sim \mathcal{N}(0, I)$, t denotes the time step of Langevin dynamics, and s is the discretization step size. The gradient term is tractable since $\frac{\partial}{\partial z}\log p_\theta(z_k|x) = \frac{\partial}{\partial z}\log p_\theta(z_k, x)$ and thus does not depend on the intractable $p_\theta(x)$. The Langevin dynamics (4) involves a gradient and a diffusion term. The first term is gradient descent $z'_{k+1} = z'_k + s\frac{\partial}{\partial z}\log p_\theta(z_k|x)$ on $\log p_\theta(z_k|x)$. If $z_k \sim p_\theta(z_k|x)$, then the distribution of z'_k will be shifted towards basins of high log-posterior. We may recover $p_\theta(z_k|x)$ by smoothing with the second term $\sqrt{2s}\epsilon_k$, which amounts to white noise diffusion and induces randomness for sampling from $p_\theta(z_k|x)$.

For small step size s, the marginal distribution of z_k will converge to $p_\theta(z|x)$ as $k \to \infty$ regardless of the initial distribution of z_0 (Cover and Thomas, 2006). More specifically, let $q_k(z)$ be the marginal distribution of z_k of the Langevin dynamics, then $\text{KL}(q_k(z)\|p_\theta(z|x))$ decreases monotonically to 0, that is, by increasing k, we reduce $\text{KL}(q_k(z)\|p_\theta(z|x))$.

Finally, the MLE learning can be accomplished by gradient descent. Each learning iteration updates θ by

$$\theta_{t+1} = \theta_t + \eta_t\frac{1}{n}\sum_{i=1}^{n}\text{E}_{p_{\theta_t}(z_i|x_i)}\left[\frac{\partial}{\partial\theta}\log p_\theta(x_i, z_i)\mid_{\theta=\theta_t}\right], \tag{5}$$

where η_t is the step size or learning rate, and $\text{E}_{p_{\theta_t}(z_i|x_i)}$ can be approximated by Monte Carlo sampling from $p_{\theta_t}(z_i|x_i)$.

2.3 Learning with short run inference dynamics

It is computationally impractical to run long Markov chains from $p_\theta(z|x)$ as the gradient term in (4) requires back-propagation through the model underlying $p_\theta(x|z)$. Earlier work (Han et al., 2017) recruits persistent Markov chains (Tieleman, 2008) $\{(z_i, x_i), i = 1, \ldots, n\}$ such that for each observed example x_i a latent code z_i is updated for a few steps in each learning iteration and the chains are maintained throughout the learning procedure. This method leads to inconsistent sampling procedures while training and evaluating the model, since persistent Markov chains for evaluation data are not available. Moreover, estimation of the log-likelihood has to resort to means such as annealed importance sampling (Neal, 2001).

Instead, we adopt short run MCMC (Nijkamp et al., 2019) in which we approximately sample from the posterior distribution of the latent variable. We thus propose the following short run inference dynamics, with a fixed small number of steps

K (e.g., $K = 20$),

$$z_0 \sim p(z), \tag{6}$$

$$z_{k+1} = z_k + s\frac{\partial}{\partial z}\log p_\theta(z_k|x) + \sqrt{2s}\epsilon_k, \tag{7}$$

where $k = 1, ..., K$ and $p(z)$ is the prior distribution of z. Initializing $z_0 \sim p(z) = \mathcal{N}(0, I)$, we perform K steps of Langevin with step size s.

Finally, the learning procedure updates θ by

$$\theta_{t+1} = \theta_t + \eta_t \frac{1}{n}\sum_{i=1}^{n} \mathrm{E}_{q_{s,\theta_t}(z_i|x_i)}\left[\frac{\partial}{\partial\theta}\log p_\theta(x_i, z_i)\ |_{\theta=\theta_t}\right], \tag{8}$$

(9)

where η_t is the learning rate, $\mathrm{E}_{q_{\theta_t}(z_i|x_i)}$ can be approximated by samples drawn from $q_{\theta_t}(z_i|x_i)$ using (7). Compared to MLE learning algorithm (5), we replace $p_{\theta_t}(z|x)$ by $q_{s,\theta_t}(z|x)$. Moreover, we may update the step size s of (7), which we will elaborate in the following.

2.4 Theoretical understanding

Given θ_t, the updating equation (9) is a one step gradient ascent on

$$Q_s(\theta) = \frac{1}{n}\sum_{i=1}^{n} \mathrm{E}_{q_{s,\theta_t}(z_i|x_i)}\left[\log p_\theta(x_i, z_i)\right]. \tag{10}$$

Compared to the log-likelihood in MLE learning, $L(\theta) = \frac{1}{n}\sum_{i=1}^{n}\log p_\theta(x)$, we have

$$\begin{aligned} Q_s(\theta) &= L(\theta) + \frac{1}{n}\sum_{i=1}^{n} \mathrm{E}_{q_{s,\theta_t}(z_i|x_i)}\left[\log p_\theta(z_i|x_i)\right] \\ &= L(\theta) - \frac{1}{n}\sum_{i=1}^{n} \mathrm{KL}(q_{s,\theta_t}(z_i|x_i)\|p_\theta(z_i|x_i)) \\ &+ \frac{1}{n}\sum_{i=1}^{n} \mathrm{E}_{q_{s,\theta_t}(z_i|x_i)}[\log q_{s,\theta_t}(z_i|x_i)]. \end{aligned} \tag{11}$$

Since the last term has nothing to do with θ, gradient ascent on $Q_s(\theta)$ is equivalent to gradient ascent of

$$\tilde{Q}_s(\theta) = L(\theta) - \frac{1}{n}\sum_{i=1}^{n} \mathrm{KL}(q_{s,\theta_t}(z_i|x_i)\|p_\theta(z_i|x_i)), \tag{12}$$

which is a perturbation or a variational lower bound of log-likelihood $L(\theta)$.

The fixed point of the learning algorithm (9) solves the following estimating equation:

$$\frac{1}{n}\sum_{i=1}^{n} \mathrm{E}_{q_{s,\theta}(z_i|x_i)}\left[\frac{\partial}{\partial\theta}\log p_\theta(x_i, z_i)\right] = 0. \tag{13}$$

If we approximate $\mathrm{E}_{q_{s,\theta_t}(z_i|x_i)}$ by Monte Carlo samples from $q_{s,\theta_t}(z_i|x_i)$, then the learning algorithm becomes Robbins-Monro algorithm for stochastic approximation (Robbins and Monro, 1951), whose convergence to the fixed point follows from regular conditions of Robbins-Monro. The estimating equation (9) is a perturbation of the maximum likelihood estimating equation $\frac{1}{n}\sum_{i=1}^{n} \mathrm{E}_{p_\theta(z_i|x_i)}\left[\frac{\partial}{\partial\theta}\log p_\theta(x_i, z_i)\right] = 0$.

2.5 Optimizing step size

We can optimize the step size s by maximizing $\tilde{Q}_s(\theta)$ defined in equation (12), which is equivalent to minimizing the KL divergence between the short-run-dynamics-induced posterior and the true posterior since the first term $L(\theta)$ does not involve s. $\tilde{Q}_s(\theta)$ involves the entropy of $q_{s,\theta_t}(z_i|x_i)$. We provide the details of its computation in the supplementary materials. The step size optimization can be done by grid search or stochastic gradient descent. In this work, we optimize the step size s with grid search guided by maximizing $\tilde{Q}_s(\theta)$.

2.6 Algorithm

The learning procedure is summarized in Algorithm 1. Note that we only optimize s every T_s iterations, so that computational cost is negligible.

Algorithm 1: Learning with SRI.

input : Learning iterations T, step size interval T_s, learning rate η, initial weights θ_0, observed examples $\{x_i\}_{i=1}^n$, batch size m, number of steps K, initial step size s.

output : Weights θ_{T+1}.

for $t = 0 : T$ **do**

1. Draw observed examples $\{x_i\}_{i=1}^m$.
2. Draw latent vectors $\{z_{i,0} \sim p(z)\}_{i=1}^m$.
3. Infer $\{z_{i,K}\}_{i=1}^m$ by K-steps of dynamics (7) with step size s.
4. Update θ according to (9).
5. Every T_s iterations, update s.

2.7 Log-likelihood computation

Unlike traditional MCMC, short run inference enables the computation of the marginal log-likelihood $\log p(x)$[1],

$$\begin{aligned}\log p_\theta(x) &= \log \int p_\theta(x, z)dz \\ &= \log \int \frac{p_\theta(x|z)p(z)}{q_k(z)} q_k(z)dz \\ &= \log E_{q_k(z)}\left[\frac{p_\theta(x|z)p(z)}{q_k(z)}\right]. \quad (14)\end{aligned}$$

Then,

$$\begin{aligned}&E_{p_{\text{data}}}\left[\log \frac{1}{M}\sum_{i=1}^{M} \frac{p_\theta(x|z_i)p(z_i)}{q_k(z_i|x)}\right] \\ &= E_{p_{\text{data}}}\Bigg[\log \sum_{i=1}^{M} \exp\left[\log p_\theta(x|z_i)\right. \\ &\left. + \log p(z_i) - \log q_k(z_i|x)\right] - \log M\Bigg]. \quad (15)\end{aligned}$$

While most terms in (15) are readily available, $\log q_k(z_i|x)$ requires special treatment. We may rewrite the dynamics (7) in the form of

$$z_0 \sim p(z), \quad z_k = R_k(z_0) \quad (16)$$

where R_k is defined by a k-step Langevin dynamics. Let the distribution of z_k be denoted $q_k(z)$. Then, by change of variable theorem,

$$z_k \sim q_k(z), \quad (17)$$

$$q_k(z) = p(R_k^{-1}(z))|\det(dR_k^{-1}(z)/dz)|. \quad (18)$$

Instead of inverting R_k, we draw $z_0 \sim p(z)$ and compute the log determinant of the Jacobian $dR_k(z_0)/dz_0$. See more details in the supplementary.

3 Related Work

Variational inference. VAE (Kingma and Welling, 2014; Bowman et al., 2016) is a prominent method for learning generative models. Due to the autoregressive nature of the decoder, a naive application of VAE to text data results in posterior collapse. Following work makes extensive efforts to alleviate this issue (Fu et al., 2019; Yang et al., 2017; He et al., 2019; Li et al., 2019; Kim et al., 2018; Pelsmaeker and Aziz, 2020; Dieng and Paisley, 2019). Among them SA-VAE developed by Kim et al. (2018) is mostly related to our work. They propose SA-VAE where initial variational parameters obtained from the inference model are further refined by running a small number of gradient updates (e.g., 20) guided by the ELBO. In our work, instead of relying on a parameteric varational distribution, we run a few gradient updates on the log-posterior of the latent variable with initialization from the prior distribution to draw samples directly. Thus, there is no need to design and tune an extra inference model, which is highly non-trivial considering that posterior collapse occurs easily in VAE training.

Alternating back-propagation. Han et al. (2017) propose to learn generative models for images by maximum likelihood, where the learning algorithm iterates over two steps: (i) inferring the latent variable by sampling from its posterior distribution with Langevin dynamics; (ii) updating the model parameters based on the inferred latent codes. In the training stage, in step (i), the Langevin dynamics is initialized from the latent codes inferred in the last epoch, which is called persistent chain in the literature (Tieleman, 2008). In contrast, the short run dynamics always initializes the gradient descent updates from the prior noise distribution. Data-independent initialization renders the dynamics in training and testing consistent.

Short run MCMC. Nijkamp et al. (2019) introduces short run MCMC as a learned sampling dynamics guided by an energy-based model. It shares the same theoretical underpinning as early work of using stochastic gradient Langevin dynamics to learn mixture of Gaussians and logistic regression for large-scale data (Welling and Teh, 2011). Our short run inference method for learning latent variable models for text is inspired by these works.

4 Experiments

We apply our method to train latent variable models on text datasets. The dimension of the latent variable is 32 in all experiments. The generator is implemented with a one-layer uni-directional LSTM (Hochreiter and Schmidhuber, 1997). The number of hidden units and word embedding size of the LSTM vary among datasets to closely follow the experimental setup in recent work (Fu et al.,

[1]Note that its Monte Carlo estimator is biased but the bias is diminishing with a large sample size.

	PPL	Recon	AU	KL
PTB				
LSTM-LM	100.47	-	-	-
Anneal-VAE	101.40	101.28	0	0.00
Cyclical-VAE	108.97	101.85	5	1.37
Lagging-VAE	99.83	100.26	4	0.93
SA-VAE	100.39	100.97	5	1.86
FBP-VAE	99.62	98.52	3	2.95
FB-VAE	96.35	94.52	32	8.15
Ours	**94.26**	**91.14**	**32**	**10.13**
SNLI				
LSTM-LM	21.44	-	-	-
Anneal-VAE	21.50	31.66	2	1.42
Cyclical-VAE	21.62	30.89	4	2.36
Lagging-VAE	**21.16**	31.53	5	1.42
SA-VAE	21.49	30.12	5	2.34
FBP-VAE	21.46	31.04	3	2.12
FB-VAE	22.00	23.36	32	8.48
Ours	21.21	**22.24**	**32**	**10.02**
Yahoo				
LSTM-LM	60.75	-	-	-
Anneal-VAE	61.52	329.10	0	0.00
Cyclical-VAE	66.93	333.80	0	2.83
Lagging-VAE	59.77	322.70	15	5.70
SA-VAE	63.92	327.27	17	7.23
FBP-VAE	62.88	328.13	2	3.06
FB-VAE	59.51	315.31	32	15.02
Ours	**57.05**	**311.23**	**32**	**16.19**

Table 1: Language modeling results on PTB, SNLI, and Yahoo test set.

FB-VAE
a man with a cane is walking down the street .
a man with a cane is walking down the street .
a man in a blue shirt is eating food .
people are eating food .
people walk in a city .
people are outside in a city .
Ours
there is a boy skating down a small street .
there is a child walking in the snow .
the man is riding a horse through the snow .
the man is riding a boat .
the biker is looking at the lake .
the person is looking at a country .

Table 2: Comparison on interpolation. Sentence samples greedily decoded from linear interpolation between samples from the Gaussian prior with FB-VAE (Top) and SRI-trained generative model (Bottom).

2019; Li et al., 2019). The number steps of the short run dynamics is 20 for all experiments [2]. The sample from the short run dynamics is used to predict the initial hidden state of the LSTM. It is also concatenated with the word embeddings and then fed to the LSTM as input at each time step.

The short run inference is more computationally costly than the vanilla VAE and has comparable training cost as some improved versions of VAE. The number of inner steps of SRI (20 steps) is about the same as that of SA-VAE and Lagging-VAE. In training, SRI has faster convergence than SA-VAE and comparable convergence as Lagging-VAE in our experiments. In inference, our sampling-based approach is slower than amortized inference. Our method trades a feasible computational cost for accurate inference whose empirical performance is presented in the following experiments.

[2] $K = 10$ steps led to posterior collapse. We observed a slight improvement in model performance if K was increased from 20 to 40 and no improvement from 40 to 60.

4.1 Language Modeling

We evaluate our method on language modeling with the Penn Tree Bank (PTB) (Marcus et al., 1993), Yahoo (Yang et al., 2017), and a down-sampled version of the Stanford Natural Language Inference (SNLI) corpus (Bowman et al., 2015) as preprocessed in (Li et al., 2019). Ideally, a language model with latent variable would be expected to make use of the latent space and accurately model the data distribution. To measure the utilization of the latent space, three quantitatively metrics are often considered in prior work (Bowman et al., 2016; Li et al., 2019; Fu et al., 2019; Kim et al., 2018): reconstruction error (Recon), number of active units (AU), the magnitude of KL. Reconstruction error is the negative log-likelihood of the observed data evaluated under the posterior, $\mathrm{E}_{q(z|x)}[-\log p_\theta(z|x)]$. A latent dimension is considered active if its distribution changes depending on the observations. Following Burda et al. (2016), a latent dimension is defined to be active if $\mathrm{Cov}_x(\mathrm{E}_{z\sim q(z|x)}[z]) > 10^{-2}$. Perplexity (PPL) based on the marginal log-likelihood of x is adopted to measure how accurately the model captures the data. The marginal log-likelihood is estimated with importance sampling with z sam-

there is a crowd of people in the city .
a man rubbing a dirty face .
a couple was waiting to cross the street in a grocery store .
the little girl is drinking water .
a group of boys are playing in the fountain
five asian teenagers are peforming a dance routine for a volunteer organization .
a white-haired man is in front of a building playing music .
construction workers sit at a african courtyard .
a jewish man wearing white garb , playing a guitar , with a lazy look on him .
a young man in a brown checkered shirt sings down on the floor while playing with the on a hot day .

Table 3: Comparison on the generated sentences. Sentence samples generated from the Gaussian prior by FB-VAE (Top) and SRI-trained generative model (Bottom).

k	1	2	3	4
AE	**26.05**	40.46	52.77	63.07
Anneal-VAE	32.20	32.65	33.12	33.39
Cyclical-VAE	31.83	32.87	33.73	34.38
Lagging-VAE	31.78	31.99	32.21	32.32
SA-VAE	31.63	31.82	32.15	32.46
FBP-VAE	29.93	32.59	34.90	36.77
FB-VAE	27.92	29.12	**30.03**	30.85
Ours	27.12	**28.66**	30.21	**30.46**

Table 4: Noisy reconstruction loss on SNLI. k is the number of word swaps performed on the original sentences.

Number of Labels	0	100	$1k$	$10k$
AE	53.1	78.8	83.7	84.1
Anneal-VAE	56.3	59.2	62.3	65.8
Cyclical-VAE	59.9	78.6	82.7	83.2
Lagging-VAE	63.6	65.8	74.2	80.5
SA-VAE	62.6	69.3	78.8	81.4
FBP-VAE	60.9	74.8	76.9	81.1
FB-VAE	67.5	84.9	89.5	90.6
Ours	**73.3**	**85.8**	**89.6**	**90.8**

Table 5: Accuracy on Yelp of unsupervised and semi-supervised classification as a function of the number of labeled example during training.

ples from trained short run dynamics as importance samples.

Besides the standard LM and the vanilla VAE with KL weight annealing, VAEs with recent state-of-the-art training techniques, Cyclical-VAE (Fu et al., 2019), Lagging-VAE (He et al., 2019), SA-VAE (Kim et al., 2018), FBP-VAE and FB-VAE (Li et al., 2019) are also included for comparison. The results are displayed in Table 1. In terms of PPL, our method outperforms all the baselines on the PTB and Yahoo datasets, while does slightly worse than Lagging-VAE and performs better than other baselines on the SNLI. This indicates the model trained with our method is able to accurately model the data distribution. On the other hand, our method yields the lowest reconstruction error and the highest KL with all latent dimension active on all three datasets, exposing the active use of the latent space. Taken together, these results suggest that the model trained with short run dynamics are balanced on modeling the data and utilizing the latent space.

Figure 1 displays a t-SNE plot of the SRI-induced aggregate posterior $E_{p_{\text{data}}} q(z|x)$ and its marginal density of each dimension. The t-SNE plot demonstrates the SRI-induced aggregate posterior is multi-modal and the marginal densities are uni-modal but clearly deviates from the zero-centered standard Gaussian prior. These visualizations demonstrate that the aggregate posterior in our model is clearly different the isotropic Gaussian prior[3] and thus our

[3]Ideally $E_{p_{\text{data}}} q(z|x) = p(z)$ since $\int_x p_{\text{data}}(x) q(z|x) = \int_x p(z) p_\theta(x|z)$. However the generative model might not be able to induce such a model posterior. The mismatch might indicate some form of under-regularization, similar to other approaches for mitigating posterior collapse such as FB-VAE.

model does not show a posterior collapse issue, consistent with our analysis above.

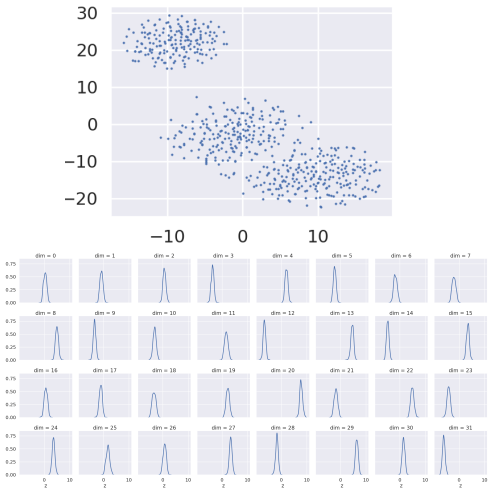

Figure 1: t-SNE plot (upper) and marginal density plot (bottom) of the SRI-induced aggregate posterior on the PTB dataset.

4.2 Latent Space Analysis

The quality of the latent space with SNLI is examined through interpolation, generation, and noisy reconstruction.

4.2.1 Interpolation

Interpolation allows us to appraise the smoothness of the latent space. In particular, two samples z_1 and z_2 are drawn from the prior. We linearly interpolate between them and then decode the interpolated points. FB-VAE (Li et al., 2019) is considered as the SOTA text VAE that mitigates posterior collapse. Due to space limit, we only include this method for comparison in interpolation and generation experiments. Table 2 shows the decoded samples. Although the interpolated sentences by FB-VAE appears smooth, the first two sentences are repetitive. In comparison, the decoded sentences from our model transition more smoothly. While the interpolated sentences from our model are diverse, their syntactic properties and topic information remain consistent in neighborhoods along the path, exposing a smooth latent space.

4.2.2 Generation

We sample from the prior distribution and decode the sentences in a greedy manner. Table 3 displays the samples from our model and FB-VAE. It appears that samples from both models are grammatically correct and semantically meaningful in general. FB-VAE samples nevertheless show more local grammar errors. More generated samples are given in the supplementary.

4.2.3 Noisy reconstruction

Zhao et al. (2018) reasons that a latent variable model's capacity on reconstructing from noisy data reveals the smoothness of the latent space. We impose discrete noise to the data by swapping tokens in a sentence for k times, where $k = 1, 2, 3, 4$ in this experiment. The reconstruction error (negative log-likelihood) under each condition is reported in Table 4. Notice that even the AE yields the lowest reconstruction when the noise is low ($k = 1$), but its performance deteriorates quickly as the noise level increases, implying that the latent space of AE is not smooth. In contrast, other models with regularization on the latent space do not exhibit drastic decline in reconstruction performance with increasing noise level. Furthermore, the model trained with our method demonstrates reconstruction either outperforming other methods or comparable to the best, revealing that the model trained with SRI has a smooth latent space.

4.3 Classification

The latent space of a well-learned latent variable model should capture highly informative features such that data points cluster into meaningful groups in the latent space. We hence further probe the latent space structure by investigating the clustering and classification performance of the SRI-inferred latent codes. Following prior work (Fu et al., 2019; Li et al., 2019), we utilize the Yelp sentiment dataset as preprocessed in Shen et al. (2017). We train a Gaussian mixture for clustering (zero labels) and a SVM with 100, 1000, or 10,000 number of labels. The results are displayed in Table 5. Our method consistently improves over VAE approaches and AE. The improvement is especially clear in the zero-shot setting and small data regime (0 and 100 labels), revealing a well-structured latent space learned by SRI.

5 Conclusion

This work proposes to use short run inference dynamics to infer latent variables in text generative

models. SRI dynamics is always initialized from the prior distribution of the latent variable and then performs a small number (e.g., 20) of Langevin dynamics updates guided by the posterior distribution. This simple and automatic inference method induces a good approximate posterior and provides good latent code.

The model trained with SRI accurately models the text data compared to strong language model and generative model baselines and shows no sign of posterior collapse, which is non-trivial to avoid and several remedies have been proposed for in prior art. Moreover, the learned space is smooth and captures rich representations of the sentences.

Acknowledgement

We thank the reviewers for their insightful comments and suggestions. The work is supported by NSF DMS-2015577.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany. Association for Computational Linguistics.

Yuri Burda, Roger B. Grosse, and Ruslan Salakhutdinov. 2016. Importance weighted autoencoders. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Tianqi Chen, Emily B. Fox, and Carlos Guestrin. 2014. Stochastic gradient hamiltonian monte carlo. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 1683–1691.

Thomas M. Cover and Joy A. Thomas. 2006. *Elements of information theory (2. ed.)*. Wiley.

Chris Cremer, Xuechen Li, and David Duvenaud. 2018. Inference suboptimality in variational autoencoders. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pages 1086–1094.

Adji B Dieng and John Paisley. 2019. Reweighted expectation maximization. *arXiv preprint arXiv:1906.05850*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*.

Hao Fu, Chunyuan Li, Xiaodong Liu, Jianfeng Gao, Asli Çelikyilmaz, and Lawrence Carin. 2019. Cyclical annealing schedule: A simple approach to mitigating KL vanishing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 240–250.

Tian Han, Yang Lu, Song-Chun Zhu, and Ying Nian Wu. 2017. Alternating back-propagation for generator network. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 1976–1984.

Junxian He, Daniel Spokoyny, Graham Neubig, and Taylor Berg-Kirkpatrick. 2019. Lagging inference networks and posterior collapse in variational autoencoders. In *Proceedings of ICLR*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Yoon Kim, Sam Wiseman, Andrew C Miller, David Sontag, and Alexander M Rush. 2018. Semi-amortized variational autoencoders. In *International Conference on Machine Learning*.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Paul Langevin. 1908. *On the theory of Brownian motion*.

Bohan Li, Junxian He, Graham Neubig, Taylor Berg-Kirkpatrick, and Yiming Yang. 2019. A surprisingly effective fix for deep latent variable modeling of text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3601–3612, Hong Kong, China. Association for Computational Linguistics.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Comput. Linguist.*, 19(2):313–330.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*.

Radford M. Neal. 2001. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139.

Radford M Neal. 2011. MCMC using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2.

Erik Nijkamp, Mitch Hill, Song-Chun Zhu, and Ying Nian Wu. 2019. Learning non-convergent non-persistent short-run MCMC toward energy-based model. *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, Canada*.

Tom Pelsmaeker and Wilker Aziz. 2020. Effective estimation of deep generative language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7220–7236, Online. Association for Computational Linguistics.

Herbert Robbins and Sutton Monro. 1951. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Advances in neural information processing systems*, pages 6830–6841.

Tijmen Tieleman. 2008. Training restricted boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th international conference on Machine learning*, pages 1064–1071. ACM.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Max Welling and Yee W Teh. 2011. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688.

Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. 2017. Improved variational autoencoders for text modeling using dilated convolutions. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 3881–3890.

Junbo Zhao, Yoon Kim, Kelly Zhang, Alexander Rush, and Yann LeCun. 2018. Adversarially regularized autoencoders. In *International Conference on Machine Learning*, pages 5897–5906.