A Tale of Two Latent Flows: Learning Latent Space Normalizing Flow with Short-Run Langevin Flow for Approximate Inference

Jianwen Xie, Yaxuan Zhu, Yifei Xu, Dingcheng Li, Ping Li

Cognitive Computing Lab
Baidu Research
10900 NE 8th St. Bellevue, WA 98004, USA
{jianwen.kenny, yaxuanzhu.alvin, fei960922, dingchengl, pingli98}@gmail.com

Abstract

We study a normalizing flow in the latent space of a top-down generator model, in which the normalizing flow model plays the role of the informative prior model of the generator. We propose to jointly learn the latent space normalizing flow prior model and the top-down generator model by a Markov chain Monte Carlo (MCMC)-based maximum likelihood algorithm, where a short-run Langevin sampling from the intractable posterior distribution is performed to infer the latent variables for each observed example, so that the parameters of the normalizing flow prior and the generator can be updated with the inferred latent variables. We show that, under the scenario of non-convergent short-run MCMC, the finite step Langevin dynamics is a flow-like approximate inference model and the learning objective actually follows the perturbation of the maximum likelihood estimation (MLE). We further point out that the learning framework seeks to (i) match the latent space normalizing flow and the aggregated posterior produced by the short-run Langevin flow, and (ii) bias the model from MLE such that the short-run Langevin flow inference is close to the true posterior. Empirical results of extensive experiments validate the effectiveness of the proposed latent space normalizing flow model in the tasks of image generation, image reconstruction, anomaly detection, supervised image inpainting and unsupervised image recovery.

1 Introduction

A flow-based model (Dinh, Krueger, and Bengio 2015; Dinh, Sohl-Dickstein, and Bengio 2016; Kingma and Dhariwal 2018), also known as normalizing flow, represents a data distribution by a non-linear invertible transformation of a simple distribution, e.g., Gaussian distribution. Thanks to the special design of the computation layers of the transformation, the flow-based model defines a normalized probability distribution explicitly, making it convenient for learning, sampling and inference. Specifically, its learning can be handily accomplished by the maximum likelihood estimation with the tractable density function, its sampling can be efficiently achieved by a direct mapping of samples drawn from a Gaussian distribution, and its inference can be easily realized by the inverse transformation of data. Therefore, flow-based models have been widely used for direct modeling of various types of data, such as images (Kingma

and Dhariwal 2018), videos (Kumar et al. 2020), point clouds (Yang et al. 2019), audio (Ping et al. 2020) etc.

In spite of the computational convenience, the expressivity of a normalizing flow is constrained by the unnatural design of the invertible transformation. Increasing the model expressivity typically requires significantly enlarging the network structure, thus leading to difficulty in constructing the network architecture and optimizing the network parameters. In order to foster the strengths and circumvent the shortcomings of the normalizing flow, we propose to learn a normalizing flow in the low-dimensional latent space of observed data. The resulting generative model becomes a normalizing flow standing on top of a top-down generator network. The latent space normalizing flow model can not only reserve the computational merit, but also focus on capturing regularities and rules of the data in the continuous latent space without having to care about the data type and data complexity. Also, the learning and the design of a normalizing flow in the latent space is much easier than those in the data space due to the low dimensionality of the latent space. Typically, it can be parameterized by a small invertible multi-layer perceptron that consists of several layers.

Besides, the proposed latent space normalizing flow can be considered an informative prior distribution of the latent variables in the top-down generator model. Compared to the commonly used isotropic Gaussian (or uniform) non-informative prior in the conventional generator model, the informative prior represented by a trainable normalizing flow can significantly increase the expressive power of the entire generator. The normalizing flow model is a good fit for the role of prior distribution in the deep latent variable framework in the sense that (i) the normalizing-flow-based prior can be jointly trained with the top-down network due to the tractability of the prior probability density and (ii) the data generation from the learned model can be explicit and fast because of the ease of the prior sampling. We call our model the latent space flow-based prior model (LFBM).

In this work, we propose a novel and principled way to jointly train the flow-based prior model and the generator via maximum likelihood estimation (MLE). The training process alternates the inference step and the learning step. Each inference step involves Markov chain Monte Carlo (MCMC) sampling of the posterior distribution, which is guided by both the flow-based prior distribution and the top-down gen-

erator, for the sake of inferring latent variables of each observed example, while, in each learning step, the parameters of the flow-based prior is updated by maximizing the likelihood of the inferred latent variables and the parameters of the generator are updated by minimizing the regression error between the inferred latent variables and the corresponding observed examples. As to the MCMC inference, we adopt a short-run Langevin dynamics, i.e., a finite-step Langevin dynamics initialized with a fixed Gaussian distribution. Even though the short-run MCMC is an approximate inference due to the non-convergence, the normalizing flow prior can be adjusted based on the approximately inferred latent variables to mend the discrepancy between the aggregate posterior distribution and the prior distribution. We show that the learning algorithm based on short-run MCMC inference is a perturbation of MLE. In contrast to the latent space energy-based prior model (LEBM) (Nijkamp et al. 2020), in which short-run MCMCs are applied to not only the posterior sampling but also the EBM prior sampling, our model is able to perform an exact prior sampling via ancestral sampling and thus less biased from MLE than the LEBM. Compared to the baseline that trains the same LFBM along with an extra bottom-up inference model (with additional parameters) for variational auto-encoders (VAEs) (Kingma and Welling 2014), our framework with a Langevin flow for approximate inference exhibits superiorities, e.g., better empirical performance, less model parameters, and free of inference network design. Table 1 presents an analytical comparison of different latent variable models using various combinations of prior and posterior components. This comparison can clearly show the advantage of our proposed model and the motivation of this work. Specifically, this paper is motivated by proposing a practical, design-friendly and lightweighted latent variable framework with informative prior.

We summarize the main contributions of our paper as follows: (1) We propose a simple but novel deep generative model where a latent space normalizing flow, which serves as a prior, stands on a single top-down generator network. (2) We propose a natural and principled maximum likelihood learning algorithm to jointly train the latent space normalizing flow and the top-down network with MCMC-based inference over latent variables. (3) We also propose to use short-run Langevin flow as an approximate inference for efficient training. (4) We provide theoretical understanding of the proposed learning framework. (5) We provide extensive and strong experimental results on different aspects, including image synthesis, inference, reconstruction, inpainting and recovery, to validate the effectiveness of the proposed models and learning algorithms.

2   Related Work

Data Space Flow-based Models and Latent Space Flow-based Models.   This work builds upon the techniques of normalizing flows, which have been widely applied to data space for representing high-dimensional data distributions directly (Dinh, Krueger, and Bengio 2015; Dinh, Sohl-Dickstein, and Bengio 2016; Kingma and Dhariwal 2018; Yang et al. 2019; Ping et al. 2020; Kumar et al. 2020; Shi et al. 2020). This work moves the flow-based models from

data space to latent space for representing low-dimensional latent manifolds from which the observed data are generated. Some prior works, such as Chen et al. (2017); Huang et al. (2017); Ma, Zhou, and Hovy (2019); Xiao, Yan, and Amit (2019); Ding and Gimpel (2021), study flow-based priors in VAE scenarios. Our paper mainly focus on maximum likelihood learning of flow-based priors with short-run Langevin MCMC inference.

Top-down Models with Implicit Density and Explicit Density.   A top-down model is a nonlinear generalization of the factor analysis model (Rubin and Thayer 1982) and its density is generally implicit and intractable. Likelihood-based training of a top-down generator model typically relies on either the Markov chain Monte Carlo sampling that evaluates the posterior distribution, such as Han et al. (2017); Xie et al. (2019) or the variational inference that approximates the posterior distribution, such as Kingma and Welling (2014). A normalizing flow is a special top-down model that defines an explicit and normalized probability density and it can be learned by maximizing the exact likelihood of the observed data through a gradient ascent algorithm. Our model also belongs to a top-down model that stacks a top-down model with explicit density on top of the other one with implicit density. Similar to other generic top-down generator models, sampling from our model can also be achieved via direct ancestral sampling. However, the flow-based model at the top serves as a flexible and tractable prior distribution for the top-down model at the bottom.

Latent Space Energy-based Prior.   Recently, Pang et al. (2020, 2021); Pang and Wu (2021); Zhang et al. (2021) propose to use deep energy-based models (EBM) (Xie et al. 2016) to serve as flexible priors for top-down generation models. Because an EBM prior defines an unnormalized density, drawing samples from such a prior requires iterative MCMC sampling, thus bringing in inconveniences during the training and sampling of the whole model. Given the inferred latent variables, the update of the latent space EBM prior involves a non-convergent short-run MCMC that samples from the prior, which leads to a biased estimation of the EBM prior distribution (Nijkamp et al. 2019; Xie et al. 2022). In contrast, our normalizing-flow-based prior model doesn't have such an issue due to its properties of tractable density and fast generation.

MCMC-based Inference.   Our LFBM is related to MLE of top-down generator model with MCMC-based inference, which is also called the alternating back-propagation (ABP) learning in Han et al. (2017); Xie et al. (2019, 2020); Zhu et al. (2019); Zhang, Xie, and Barnes (2020); An, Xie, and Li (2021); Xing et al. (2022). The original ABP algorithm is designed to train a generator network with a known and simple Gaussian prior. The LEBM (Pang et al. 2020) generalizes the ABP algorithm to a framework with a trainable and unnormalized energy-based prior model, while our paper generalizes the ABP algorithm to a framework with a trainable and normalized flow-based prior model.

Variational Inference.   Most regular variational auto-encoder (VAE) frameworks employ a fixed Gaussian distri-

Methods (prior + posterior)	easy design	informative prior	fast sample	fast inference	less parameters	fast training	practical
Normalizing Flow + short-run MCMC	●	●	●	◐	●	◐	●
Normalizing Flow + long-run MCMC	●	●	●	○	●	○	○
Normalizing Flow + inference net	○	●	●	●	○	●	●
EBM + short-run MCMC	●	●	○	◐	●	◐	●
EBM + long-run MCMC	●	●	○	○	●	○	○
EBM + inference net	○	●	○	●	○	●	●
Gaussian + short-run MCMC	●	○	●	◐	●	◐	●
Gaussian + long-run MCMC	●	○	●	○	●	○	○
Gaussian + inference net	○	○	●	●	○	●	●

Table 1: A comparison of latent variable models with different priors and inference process. (● is good, ◐ is OK, and ○ is bad.)

bution as the prior and use reparameterization trick (Kingma and Welling 2014) to design the amortized inference network that outputs a Gaussian posterior. Some prior works, such as Rezende and Mohamed (2015); Kingma et al. (2016), improve the amortized inference network by using a normalizing flow that outputs a flexible posterior. Our LFBM parameterizes the prior by a normalizing flow, treats the built-in short-run Langevin MCMC as the inference model, and learns them simultaneously from the observed data by following the empirical Bayes principle.

3 Proposed Framework

3.1 Normalizing Flow in Latent Space

Let $x \in \mathbb{R}^D$ be an observed signal such as an image, and let $z \in \mathbb{R}^d$ be the latent variables of x. The joint distribution of the signal and the latent variables (x, z) is given by $p_\theta(x, z) = p_\alpha(z)p_\beta(x|z)$, where the distribution of z, i.e., $p_\alpha(z)$, is the prior model parameterized by α, and the conditional distribution of x given z, i.e., $p_\beta(x|z)$, is the top-down generation model parameterized by β. For notational convenience, let $\theta = (\alpha, \beta)$.

The top-down generaton model is a non-linear transformation of the latent variables z to generate the signal x, in which the transformation is parameterized by a neural network $g_\beta : \mathbb{R}^d \rightarrow \mathbb{R}^D$, which is similar to the decoder of a variational auto-encoder (VAE) (Kingma and Welling 2014). To be specific, $x = g_\beta(z) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2 I_D)$ is an observation residual. Thus, $p_\beta(x|z) = \mathcal{N}(g_\beta(z), \sigma^2 I_D)$, where the standard deviation σ is a hyper-parameter and assumed to be given.

For most existing top-down generative models, the prior models are assumed to be non-informative isotropic Gaussian distributions. In this paper, we propose to formulate the prior $p_\alpha(z)$ as a flow-based model (Dinh, Sohl-Dickstein, and Bengio 2016; Dinh, Krueger, and Bengio 2015; Kingma and Dhariwal 2018), which is of the form

$$z_0 \sim q_0(z_0), \ z = f_\alpha(z_0), \tag{1}$$

where $q_0(z_0)$ is a base distribution and has a simple and tractable density, such as a Gaussian white noise distribution: $q_0(z_0) = \mathcal{N}(0, I_d)$. $f_\alpha : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is an invertible or bijective function, which is a composition of a sequence of invertible transformations, i.e., $f_\alpha(z_0) = f_{\alpha_L} \circ \cdots \circ f_{\alpha_2} \circ$

$f_{\alpha_1}(z_0)$, whose inverse and logarithm of the determinants of the Jacobians can be explicitly obtained in closed form. Examples of such architectures include NICE (Dinh, Sohl-Dickstein, and Bengio 2016), RealNVP (Dinh, Krueger, and Bengio 2015), and Glow (Kingma and Dhariwal 2018).

For $l = 1, ..., L$, let $z_l = f_{\alpha_l}(z_{l-1})$ be a sequence of random variables transformed from z_0, and define $z := z_L$. According to the change-of-variable law of probabilities, $q_0(z_0)dz_0 = p_\alpha(z)dz$, the density of the flow-based prior model can be written as

$$
\begin{aligned}
p_\alpha(z) \ &= q_0(z_0)\frac{dz_0}{dz} = q_0(f_\alpha^{-1}(z)) \left| \det\left(\frac{\partial f_\alpha^{-1}(z)}{\partial z} \right) \right| \\
&= q_0(f_\alpha^{-1}(z)) \prod_{l=1}^{L} \left| \det\left(\frac{\partial z_{l-1}}{\partial z_l} \right) \right|,
\end{aligned}
\tag{2}
$$

where $f_\alpha^{-1}(z) = f_{\alpha_1}^{-1} \circ \cdots \circ f_{\alpha_{L-1}}^{-1} \circ f_{\alpha_L}^{-1}(z)$, and the determinant of the Jacobian matrix $(\partial z_{l-1}/\partial z_l)$ can be easy to compute with well-designed transformation functions in the flow-based models. Typically, choosing transformations that can lead to a triangle Jacobian matrix will simplify the computation of the determinant: $|\det(\partial z_{l-1}/\partial z_l)| = \prod |\text{diag}(\partial z_{l-1}/\partial z_l)|$, where diag() takes the diagonal elements of the Jacobian matrix. Thus, the flow-based prior model has two nice properties: (i) analytically tractable normalized density (i.e., Eq. (2)), and (ii) easy to draw samples from by using ancestral sampling (i.e., Eq. (1)).

The flow-based prior model in Eq. (1) can be regarded as a flow-based refinement or calibration of the original Gaussian prior distribution q_0, which is a widely used prior distribution in most top-down generative frameworks. There are two advantages of using flow-based prior models for generative learning: (i) This enables us to learn a flexible prior from data to capture more meaningful latent components compared to those using a simple Gaussian prior model; (ii) The original generator learns g_β to map from a fixed unimodal prior q_0 to the highly multi-modal data distribution, while in the proposed framework, the flow-based prior model corrects q_0 such that g_β can be easier to transform the calibrated distribution p_α to the data distribution.

3.2 Maximum Likelihood Learning with Langevin Inference

The top-down generative model with a flow-based prior can be trained via maximum likelihood estimation. For the train-

ing examples $\{x_i, i = 1, ..., N\}$, the observed-data log-likelihood function is given by

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \log p_\theta(x_i), \qquad (3)$$

where the marginal distribution is obtained by integrating out the latent variables z: $p_\theta(x) = \int p_\theta(x, z)dz = \int p_\alpha(z)p_\beta(x|z)dz$. Maximizing the log-likelihood function $\mathcal{L}(\theta)$ is equivalent to minimizing the Kullback-Leibler (KL) divergence between the model $p_\theta(x)$ and the data distribution $p_{\text{data}}(x)$. The gradient of $\mathcal{L}(\theta)$ is computed according to

$$\begin{aligned}
\nabla_\theta \log p_\theta(x) &= \mathbb{E}_{p_\theta(z|x)} [\nabla_\theta \log p_\theta(x, z)] \\
&= \mathbb{E}_{p_\theta(z|x)} [\nabla_\theta (\log p_\alpha(z) + \log p_\beta(x|z))],
\end{aligned} \qquad (4)$$

where the posterior distribution of z is given by $p_\theta(z|x) = p_\theta(x, z)/p_\theta(x) \propto p_\alpha(z)p_\beta(x|z)$. The inference distribution $p_\theta(z|x)$ is dependent on both the prior model α and the generation model β.

For the flow-based prior model,

$$\begin{aligned}
\log p_\alpha(z) &= \log q_0(f_\alpha^{-1}(z)) + \sum_{l=1}^{L} \log |\det(\frac{\partial z_{l-1}}{\partial z_l})| \\
&= \log q_0(z_0) + \sum_{l=1}^{L} \text{sum}(\log |\text{diag}(\frac{\partial z_{l-1}}{\partial z_l})|)
\end{aligned} \qquad (5)$$

where $\log()$ takes element-wise logarithm, and $\text{sum}()$ takes the sum over all elements in a vector. Given a datapoint z, computing its log-likelihood only need one pass of the inverse function f_α^{-1}. We define $l_\alpha(z) = \log p_\alpha(z)$ to explicitly indicate that computing the log-likelihood of z under the flow-based model is computationally tractable and can be regarded as a function of z. The learning gradient of α for a datapoint x is

$$\begin{aligned}
\nabla_\alpha \log p_\theta(x) &= \mathbb{E}_{p_\theta(z|x)} [\nabla_\alpha \log p_\alpha(z)] \\
&= \mathbb{E}_{p_\theta(z|x)} [\nabla_\alpha l_\alpha(z)].
\end{aligned} \qquad (6)$$

The update of the latent space flow-based prior model depends on the observed example x, but different from the original data space flow-based model, it treats the latent variables inferred from x as a training example and seeks to maximize the log-likelihood of the inferred latent variables. The updated $p_\alpha(z)$ will further influence the inference accuracy of $p_\theta(z|x)$.

As to the generation model, the learning gradient of β for a datapoint x is

$$\nabla_\beta \log p_\theta(x) = \mathbb{E}_{p_\theta(z|x)} [\nabla_\beta \log p_\beta(x|z)]. \qquad (7)$$

Since $p_\beta(x|z)$ is in the form of a Gaussian distribution with a mean of $g_\beta(z)$ and a standard deviation of σ, $\nabla_\beta \log p_\beta(x|z) = \nabla_\beta(-\frac{1}{2\sigma^2}||x - g_\beta(z)||^2 + \text{const}) = \frac{1}{\sigma^2}(x - g_\beta(z))\nabla_\beta g_\beta(z)$.

The learning gradient of θ in Eq.(4) is decomposed into the learning gradient of α in Eq.(6) and the learning gradient of β in Eq.(7), both of which involve the expectation with respect to the intractable posterior $p_\theta(z|x)$, which can

be approximated by drawing samples from $p_\theta(z|x)$ and then computing the Monte Carlo average. Sampling from $p_\theta(z|x)$ can be achieved by Langevin dynamics that iterates

$$\begin{aligned}
z_{(k+1)} &= z_{(k)} + \xi \nabla_z \log p_\theta(z_{(k)}|x) + \sqrt{2\xi}\epsilon_{(k)}; \\
z_{(0)} &\sim q_0(z), \epsilon_{(k)} \sim \mathcal{N}(0, I_d),
\end{aligned} \qquad (8)$$

where k indexes the Langevin time step, $\nabla_z \log p_\theta(z|x) = \nabla_z \log p_\alpha(z) + \nabla_z \log p_\beta(x|z) = \nabla_z l_\alpha(z) + \frac{1}{\sigma^2}(x - g_\beta(z))\nabla_z g_\beta(z)$, and ξ is the Langevin step size. Algorithm 1 presents the learning algorithm of LFBM. We use the Adam optimizer (Kingma and Ba 2015) to update the parameters.

3.3 Perturbation of Maximum Likelihood

The Langevin inference in Eq.(8) is practically non-mixing and non-convergent because MCMC chains from different stating points can get trapped in local modes of the posterior distribution or MCMC chains are not sufficiently long to converge. Using a short-run non-convergent MCMC, e.g., a finite-step Langevin dynamics, as a flow-like approximate inference is more computationally efficient than using a long-run MCMC and more implementationally convenient than using an amortized inference network. We use $\hat{p}_\theta(z|x)$ to denote the short-run Langevin flow distribution, which is obtained by running a K-step Langevin dynamics starting from a fixed initial distribution $q_0(z)$ toward the true posterior $p_\theta(z|x)$. Strictly speaking, Algorithm 1 with a short-run MCMC inference is a perturbation of maximum likelihood. That is, given θ_t at iteration t, the learning gradient of θ is

$$\hat{\mathcal{L}}(\theta) = \mathcal{L}(\theta) - \frac{1}{N} \sum_{i=1}^{N} \mathbb{D}_{\text{KL}}(\hat{p}_{\theta_t}(z_i|x_i)||p_\theta(z_i|x_i)), \qquad (9)$$

which is a lower bound of $\mathcal{L}(\theta)$. Theoretically, if $K \to \infty$, $\xi \to 0$, then $\mathbb{D}_{\text{KL}}(\hat{p}_\theta(z|x)||p_\theta(z|x)) \to 0$. The estimate of θ is an MLE solution. However, $\mathbb{D}_{\text{KL}}(\hat{p}_\theta(z|x)||p_\theta(z|x)) \neq 0$ in practise, the estimate of θ is biased from MLE.

In Eq.(9), the first term $\mathcal{L}(\theta)$ corresponds to MLE, i.e., we want $p_\theta(x)$ to be close to p_{data}, while the second KL-divergence term means that we want to bias θ from MLE so that $p_\theta(z|x)$ is close to $\hat{p}_{\theta_t}(z|x)$. In fact, this is not a bad thing because it means we intend to bias the model so that short-run MCMC is close to the true posterior.

The algorithm converges when learning gradients equal to zeros. The resulting estimators of $\theta = (\alpha, \beta)$ solve the following estimating equations:

$$\frac{1}{N} \sum_{i=1}^{N} \mathbb{E}_{\hat{p}_\theta(z_i|x_i)} [\nabla_\alpha \log p_\alpha(z_i)] = 0, \qquad (10)$$

$$\frac{1}{N} \sum_{i=1}^{N} \mathbb{E}_{\hat{p}_\theta(z_i|x_i)} [\nabla_\beta \log p_\beta(x_i|z_i)] = 0. \qquad (11)$$

In contrast to LFBM, the latent space energy-based prior model (LEBM) (Pang et al. 2020) adopts an unnormalized energy-based density $p_\alpha^{\text{ebm}}(z) \propto \exp(f_\alpha(z))q_0(z)$ as the prior. The MLE of LEBM requires not only MCMC inference from $p_\beta(z|x)$ but also MCMC sampling from $p_\alpha^{\text{ebm}}(z)$. The short-run MCMC inference and sampling cause the

learning algorithm of the LEBM to be a perturbation of MLE. Given θ_t, the update of θ is based on the gradient of

$$\hat{\mathcal{L}}_{\text{LEBM}}(\theta) = \mathcal{L}(\theta) - \frac{1}{N}\sum_{i=1}^{N} \mathbb{D}_{\text{KL}}(\hat{p}_{\theta_t}(z_i|x_i)||p_\theta(z_i|x_i)) \tag{12}$$
$$+ \mathbb{D}_{\text{KL}}(\hat{p}_{\alpha_t}^{\text{ebm}}(z)||p_\alpha^{\text{ebm}}(z)),$$

where $\hat{p}_\alpha^{\text{ebm}}(z)$ represents the short-run MCMC distribution. The second KL-divergence term in Eq.(12) is due to imperfect sampling from the EBM prior. Our flow-based prior $p_\alpha(z)$ is capable of exact sampling, therefore the second KL-divergence term disappears in Eq.(9). Thus, $\hat{\mathcal{L}}_{\text{LEBM}}(\theta)$ is more biased from MLE than $\hat{\mathcal{L}}(\theta)$.

Algorithm 1: Maximum likelihood learning of latent space normalizing flow model

---

Input: (1) Observed signals for training $\{x_i\}_i^N$; (2) Maximal number of learning iterations T; (3) Numbers of Langevin steps for posterior K; (4) Langevin step size ξ for the posterior; (5) Learning rates for flow-based prior model and the generation model $\{\eta_\alpha, \eta_\beta\}$.
Output: Parameters β for the generation model and α for the flow-based prior model

 1: Randomly initialize α and β
 2: **for** $t \leftarrow 1$ to T **do**
 3:     Sample a batch of observed examples $\{x_i\}_i^n$
 4:     For each x_i, sample the posterior $z_i \sim p_\theta(z|x_i)$ using K Langevin steps in Eq.(8) with a step size ξ.
 5:     Update flow-based prior by Adam optimizer with the gradient $\nabla\alpha$ in Eq.(6) and a learning rate γ_α.
 6:     Update generation model by Adam optimizer with the gradient $\nabla\beta$ in Eq.(7) and a learning rate γ_β.
 7: **end for**

3.4 Matching Normalizing Flow Prior and Aggregated Langevin Flow Posterior

We further reveal the interaction between the normalizing flow prior and the Langevin flow posterior in latent space during maximum likelihood estimation. The maximum likelihood learning minimizes the KL-divergence between the aggregated posterior and the prior. Let $\tilde{p}(x,z) = p_{\text{data}}(x)p_\theta(z|x)$, and then the aggregated posterior is $\tilde{p}(z) = \int \tilde{p}(x,z)dx = \mathbb{E}_{p_{\text{data}}(x)}[p_\theta(z|x)]$. Also, we have $\tilde{p}(x,z) = \tilde{p}(z)\tilde{p}(x|z)$. We show that

$$\begin{aligned}
&\mathbb{D}_{\text{KL}}(p_{\text{data}}(x)||p_\theta(x)) \\
=&\mathbb{D}_{\text{KL}}(p_{\text{data}}(x)p_\theta(z|x)||p_\theta(x)p_\theta(z|x)) \\
=&\mathbb{D}_{\text{KL}}(\tilde{p}(z)\tilde{p}(x|z)||p_\alpha(z)p_\beta(x|z)) \\
=&\mathbb{D}_{\text{KL}}(\tilde{p}(z)||p_\alpha(z)) + \mathbb{D}_{\text{KL}}(\tilde{p}(x|z)||p_\beta(x|z)),
\end{aligned} \tag{13}$$

which means that the maximum likelihood learning of the whole latent variable models involves a behavior that minimizes the KL divergence between the aggregated posterior and the normalizing flow in the latent space.

The above analysis is based on the ideal scenario with a long-run convergent Langevin inference. In our

framework, we assume and allow short-run Langevin sampling, and then our objective turns into maximizing the perturbation of maximum likelihood, i.e., $\log p_\theta(x) - \mathbb{D}_{\text{KL}}(\hat{p}_{\theta_t}(z|x)||p_\theta(z|x))$, or equivalently minimizing $\mathbb{D}_{\text{KL}}(p_{\text{data}}(x)||p_\theta(x)) + \mathbb{D}_{\text{KL}}(\hat{p}_{\theta_t}(z|x)||p_\theta(z|x))$. We now generalize the above analysis to the short-run non-convergent Langevin flow scenario. We show that

$$\begin{aligned}
&\mathbb{D}_{\text{KL}}(p_{\text{data}}(x)||p_\theta(x)) + \mathbb{D}_{\text{KL}}(\hat{p}_{\theta_t}(z|x)||p_\theta(z|x)) \\
=&\mathbb{D}_{\text{KL}}(p_{\text{data}}(x)\hat{p}_{\theta_t}(z|x)||p_\theta(x)p_\theta(z|x)) \\
=&\mathbb{D}_{\text{KL}}(p_{\text{data}}(x)\hat{p}_{\theta_t}(z|x)||p_\alpha(z)p_\beta(x|z)) \\
=&\mathbb{D}_{\text{KL}}(p_{\text{data}}(z)\hat{p}_{\theta_t}(x|z)||p_\alpha(z)p_\beta(x|z)) \\
=&\mathbb{D}_{\text{KL}}(p_{\text{data}}(z)||p_\alpha(z)) + \mathbb{D}_{\text{KL}}(\hat{p}_{\theta_t}(x|z)||p_\beta(x|z)),
\end{aligned} \tag{14}$$

where $p_{\text{data}}(z) = \int p_{\text{data}}(x)\hat{p}_{\theta_t}(z|x)dx$ is the aggregated posterior. $\hat{p}_{\theta_t}(x|z) = p_{\text{data}}(x)\hat{p}_{\theta_t}(z|x)/p_{\text{data}}(z)$ is the generation model based on data distribution and the inference model (i.e., short-run Langevin flow). KL-divergence between conditional distributions is understood to be averaged over the variable z being conditioned upon. In the above, we update $p_\alpha(z)$ to be close to $p_{\text{data}}(z)$. We update $p_\beta(x|z)$ to be close to $\hat{p}_{\theta_t}(x|z)$. Both $p_{\text{data}}(z)$ and $\hat{p}_{\theta_t}(x|z)$ depends on the short-run Langevin flow $\hat{p}_{\theta_t}(z|x)$.

Therefore, the learning of our model accomplishes two things: (1) Let the flow-based prior match the aggregated posterior produced by the short-run Langevin flow inference model. (2) Bias the model from MLE so that the short-run Langevin flow inference is close to the true posterior.

4 Experiments
4.1 Experiment Settings
We conduct a set of experiments to examine the effectiveness of the proposed latent space flow-based prior model in image modeling. In particular, we evaluate the model from the perspectives of synthesis, reconstruction and inference. We use SVHN (Netzer et al. 2011), CelebA (Liu et al. 2015), and CIFAR-10 (Krizhevsky 2009) image datasets for generation and reconstruction tasks, and use MNIST (LeCun et al. 1998) dataset for anomaly detection. All the inpainting and recovery experiments are conducted on CelebA face dataset. The network architectures and the hyperparameters are presented in Supplementary Material. The experiments are mainly for the sake of proof of concepts rather than state-of-the-art performance. We will compare our framework with two important baseline methods, which are (1) LFBM-VAE baseline, which trains LFBM using variational inference instead of MCMC inference, (2) LEBM baseline, which uses an energy-based model in the latent space. For fair comparison, the architecture designs of the top-down generators g_β in LEBM, LFBM-VAE and ours are the same, and the flow-based prior models f_α in LFBM-VAE and ours are also identical. The design of the inference network of LFBM-VAE follows Kingma et al. (2016), who parameterize the inference network by an encoder followed by an inverse autoregressive flow. The designed network architectures are expected to vary in different datasets due to various image sizes and pattern complexities. We use LFBM-MCMC to denote our LFBM using Langevin inference in order to distinguish from the baseline LFBM-VAE.
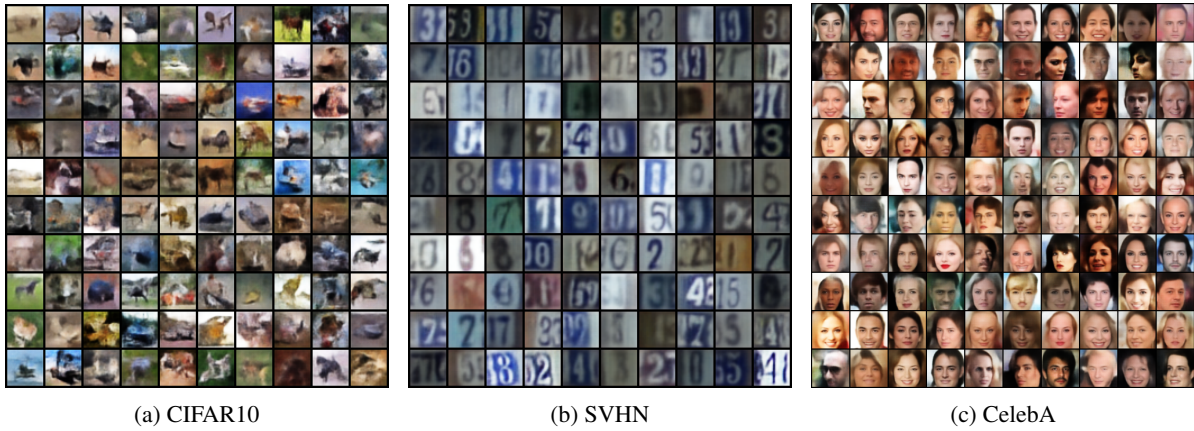
| (a) CIFAR10 | (b) SVHN | (c) CelebA |

Figure 1: Generated examples from the LFBM-MCMC models traind on the CIFAR10 (32× 32), SVHN, (32× 32) and CelebA (64× 64) datasets. The LFBM-MCMC model is trained with short-run Langevin flow as approximate inference.

4.2 Image Synthesis and Reconstruction

A well-trained latent variable generative model can be useful for generation and reconstruction. We train models on training images of SVHN, CelebA and CIFAR-10 respectively, and generate synthesized examples by first sampling latent vectors from the learned latent space normalizing flow and then transforming the vectors to image space. We calculate the Fréchet inception distance (FID) (Heusel et al. 2017) to measure the quality of the synthesized images in Table 2. We show generated samples of the learned LFBM-MCMC models in Figure 1. The models can reconstruct images by first inferring the latent vectors from the images, and then mapping the inferred latent vectors back to data space. The inference of latent variables can be achieved by the MCMC in our LFBM-MCMC. The quality of reconstruction images are measured by mean squared error (MSE) in Table 2. We mainly compare our methods with likelihood-based top-down models rather than adversarial frameworks. Specifically, we compare with traditional variational inference baselines, such as VAE (Kingma and Welling 2014) and SRI (Nijkamp et al. 2020) which use Gaussian priors and two VAE variants, 2sVAE and RAE, which learn their priors from posterior samples in the second stage. We also compare with MCMC inference frameworks, such as ABP (Han et al. 2017) and LEBM (Pang et al. 2020) whose prior distributions are fixed Gaussian distribution and energy-based model, respectively. We also compare our LFBM-MCMC with the VAE variant, i.e., the LFBM-VAE. Our models outperform other baselines in terms of MSE and FID.

4.3 Supervised Image Inpainting

We can train an LFBM-MCMC from fully-observed training images, and then use the learned model to complete the missing pixels of testing images. Let m be a matrix, which has the same number of dimension as that of an image x, with values ones indicating the visible pixels and zeros indicating the invisible ones (corrupted or occluded) of the image x, respectively. Suppose there is an incomplete image x_m with missing pixels indicated by a mask

m. With the learned flow-based prior model $p_\alpha(z)$ and the generation model $p_\beta(x|z)$, we can restore the missing pixels by first inferring the latent vectors z from x_m, and then using $g_\beta(z_m)$ to generate an image x, which is a complete version of x_m. The inference can be easily performed by Langevin dynamics that follows Eq.(8) with the gradient of a modified log-posterior $\nabla_z \log p_\theta(z|x_m, m) = \nabla_z \log p_\alpha(z) + \nabla_z \log p_\beta(x_m|z, m) = \nabla_z l_\alpha(z) + \frac{1}{\sigma^2}(m \odot (x_m - g_\beta(z)))\nabla_z g_\beta(z)$, where $\odot$ is the element-wise multiplication operator. That is, we evaluate the posterior over only the visible pixels of the images. Each testing image is occluded by a 40×40 region mask at the center. The resolution of the images are 64×64 pixels. For each testing image, our LFBM-MCMC model can generate diverse and meaningful inpainting results by using different z_0 sampled from $q_0(z)$ to initialize the Langevin chains for inference.

Figure 2 displays some qualitative results, where we compare our LFBM-MCMC model with the baseline LFBM-VAE. Both LFBM-MCMC and LFBM-VAE are trained on fully observed training images of CelebA dataset. For task of inpainting, given an incomplete image with a mask occluding pixels at the center, our LFBM-MCMC first infers the latent variables of the incomplete image by the Langevin flow, and then generate the occluded region of the image by the top-down generator taking the inferred variables as input. As to the LFBM-VAE, the learned inference model in the LFBM-VAE is hard to employ for inferring latent variables from an incomplete data, which means that even though the variational inference is computationally efficient, it is not suitable to infer latent variables from a partially observed image because the encoder can not map a portion of the image into the latent space. In order to use the LFBM-VAE for image inpainting, we abandon the learned inference network, and derive the Langevin dynamics from the learned prior and generator models in LFBM-VAE. We adopt the derived Langevin flow as in the LFBM-MCMC for inference and inpainting. As shown in Figure 2, each row illustrates one inpainting task. The first column displays the original images, and the second column shows the testing images

Models		VAE	2sVAE	RAE	SRI	SRI (L=5)	ABP	LEBM	LFBM VAE	MCMC
SVHN	MSE	0.019	0.019	0.014	0.018	0.011	-	0.008	0.005	**0.005**
	FID	46.78	42.81	40.02	44.86	35.23	49.71	29.44	24.96	**23.64**
Cifar10	MSE	0.057	0.056	0.027	-	-	0.018	0.020	0.020	**0.016**
	FID	106.37	72.90	74.16	-	-	90.30	70.15	69.70	**66.41**
CelebA	MSE	0.021	0.021	0.018	0.020	0.015	-	0.013	0.014	**0.011**
	FID	65.75	44.40	40.95	61.03	47.95	51.50	37.87	33.64	**33.64**

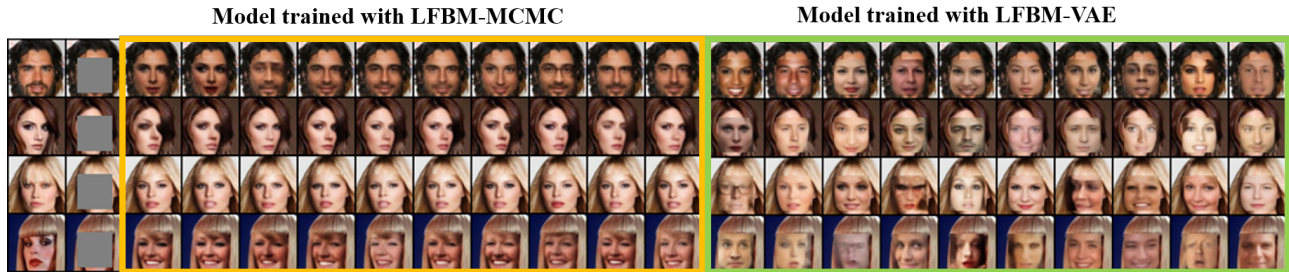Table 2: Quantitative results of image reconstruction and generation on different datasets.

Figure 2: Supervised image inpainting results on the CelebA dataset. Images in the first column are the original images. Images in the second column are the masked images to be inpainted. Images in column 3 to column 12 (yellow panel) are inpainting results using the learned LFBM-MCMC model. Images in column 13 to column 22 (green panel) are inpainting results using the trained LFBM-VAE. For each panel, different columns correspond to different initializations of the inference process.

that need to be inpainted. The yellow panel shows inpainted images by the LFBM-MCMC, while the green panel shows inpainted images by the LFBM-VAE. Different columns in each panel show different inpainting results due to the different initialization (i.e., randomness) of the inference process. Although the LFBM-VAE adopts the MCMC inference for inpainting, its results are still not as good as those by our LFBM-MCMC. Some obvious artifacts are observed in the inpainting results of the LFBM-VAE. Further, we quantitatively evaluate the image inpainting performance of both LFBM-MCMC and LFBM-VAE in Table 3. We use two metrics, i.e., FID and MSE, to measure the quality of the inpainting performance. We test the models in 10,000 images. For the MSE, we compute the per pixel difference between the inpianting result and the ground truth image within the masked region. Our model outperforms the baseline.

Method	FID	MSE
LFBM-MCMC (ours)	**0.0557**	**0.1937**
LFBM-VAE	0.0562	0.2074

Table 3: Quantitative results of image inpainting.

4.4 Anomaly Detection

We evaluate our generative model on anomaly detection of MNIST (LeCun et al. 1998) data. Given a latent variable model that is well-trained on normal examples, we can perform anomaly detection on a testing image x by firstly inferring its latent variables z and then computing the logarithm of the joint probability $\log p_\theta(x,z) = \log p_\alpha(z) +$

$\log p_\beta(x|z) = l_\alpha(z) - \frac{1}{2\sigma^2}||x - g_\beta(z)||^2 - \log \sigma\sqrt{2\pi}$ as a decision score. The score should be high for a normal example and low for an anomalous one. We can see that a normal example can be well-reconstructed by the learned generator (i.e., small reconstruction error $||x - g_\beta(z)||^2$) with a correctly inferred latent variables (i.e., high log likelihood in the learned prior distribution $l_\alpha(z)$).

Following the same experiment setup as Zenati et al. (2018a); Kumar et al. (2019); Pang et al. (2020), we treat each class of digit images in MNIST dataset as anomaly examples and the remaining 9 classes of digit images as normal examples. We train models only on the normal examples, and then test the models on both the normal and anomalous examples. To quantify the model performance, we compute the area under the precision-recall curve (AUPRC) based on the decision function $\log p_\theta(x,z)$. We report the mean and variance of AUPRC scores over 10 runs of each experiment. We show the results of our LFBM in Table 4, and compare with the related models, including the VAE (Kingma and Welling 2014), MEG (Kumar et al. 2019), BiGAN-σ (Zenati et al. 2018b), EBM-VAE (Han et al. 2020), LEBM (Pang et al. 2020) and ABP model (Han et al. 2017). From Table 4, we can find that the proposed LFBM can obtain much better results than those of other methods.

4.5 Analysis of Hyperparameters

In Figure 3(a), we show the effects of using different Langevin step sizes and numbers of steps for LFBM. We carry out experiments on SVHN dataset. We can see that the best FID score is achieved when the number of Langevin steps is around 40 and the model with 20 steps can already

Heldout Digit	1	4	5	7	9
VAE (Kingma and Welling 2014)	0.063	0.337	0.325	0.148	0.104
MEG (Kumar et al. 2019)	0.281 ± 0.035	0.401 ± 0.061	0.402 ± 0.062	0.290 ± 0.040	0.342 ± 0.034
BiGAN-σ (Zenati et al. 2018b)	0.287 ± 0.023	0.443 ± 0.029	0.514 ± 0.029	0.347 ± 0.017	0.307 ± 0.028
EBM-VAE (Han et al. 2020)	0.297 ± 0.033	0.723 ± 0.042	0.676 ± 0.041	0.490 ± 0.041	0.383 ± 0.025
LEBM (Pang et al. 2020)	0.336 ± 0.008	0.630 ± 0.017	0.619 ± 0.013	0.463 ± 0.009	0.413 ± 0.010
ABP (Han et al. 2017)	0.095 ± 0.028	0.138 ± 0.037	0.147 ± 0.026	0.138 ± 0.021	0.102 ± 0.033
LFBM (ours)	$\mathbf{0.349 \pm 0.002}$	$\mathbf{0.812 \pm 0.007}$	$\mathbf{0.823 \pm 0.009}$	$\mathbf{0.682 \pm 0.004}$	$\mathbf{0.514 \pm 0.008}$

Table 4: AUPRC scores (larger is better) for unsupervised anomaly detection. Results are averaged over 10 experiments.

give us good results. On the other hand, the optimal step sizes for Langevin flows using different numbers of steps are different. Figure 3(b) displays the effect of tuning latent size. We can see the best result can be achieved in the range from 50 to 100. Figure 3(c) shows FID scores using different depths of the flow-based prior on Cifar10. We find that increasing the depth of the prior can improve the performance.

4.6 Unsupervised Image Recovery

The LFBM can be learned from incomplete training data, e.g., images with occluded pixels. The learning algorithm updates the model parameters by maximizing the likelihood of the visible pixels in training images. This also belongs to unsupervised image inpainting, which is essentially different from the task of supervised image inpainting shown in Section 4.3. Traditional VAE-based frameworks, such as LFBM-VAE, are incapable of learning from incomplete data. To demonstrate this ability of the LFBM, we experiment on 10,000 occluded images that we create. They are selected from the CelebA dataset and occluded with different types of masks. A mask is randomly placed in each image. Two types of masks are designed, including single region mask and salt-and-pepper mask. We specify three different sizes of single regions, e.g., 20×20, 30×30, and 40×40, and three different occlusion percentages of salt-and-pepper masks, e.g., 30%, 50%, and 70%. We compare our LFBM with a fixed Gaussian prior and an energy-based prior learned by MCMC inference. We can evaluate the performance from two aspects: (i) Recovery quality: The incomplete training images are gradually recovered as the learning algorithm proceeds. MSE between recovered images and the corresponding ground truth images at masked regions are calculated; (ii) Generation quality: We use FID to measure the visual quality of the synthesized images, which are generated by models learned from incomplete data. Table 5 and Table 6 show comparisons of our method with the baselines using EBM prior and Gaussian prior in terms of MSE and FID, respectively. Figure 4 shows the recovery results for learning from incomplete training images of the CelebA dataset. We further evaluate the generation capacities of the models trained in this scenario. We present qualitative results in Figure 5, which shows the randomly generated images by the models learned from incomplete images with different occlusion levels. We can see that frameworks using Gaussian priors have the worst generated results. Our LFBM model using Langevin flow as an infer-

ence process can generate more realistic images than other baselines in the task of unsupervised learning.

Salt and pepper mask			
Occ %	30%	50%	70%
flow (ours)	**0.0244**	**0.0317**	**0.0464**
EBM	0.0256	0.0319	0.0465
Gaussian	0.0259	0.0326	0.0472
Single region mask			
mask size	20×20	30×30	40×40
flow (ours)	0.0420	0.0587	**0.0864**
EBM	0.0429	0.0684	0.0957
Gaussian	**0.0404**	**0.0572**	0.0918

Table 5: MSEs of methods with different priors (e.g., flow, EBM and Gaussian) in unsupervised image recovery.

Salt and pepper mask			
Occ %	30%	50%	70%
flow (ours)	**46.2**	**59.14**	**86.77**
EBM	52.78	61.91	88.27
Gaussian	153.01	156.71	172.77
Single region mask			
mask size	20×20	30×30	40×40
flow (ours)	**42.39**	**47.52**	**72.47**
EBM	49.16	51.59	77.39
Gaussian	150.95	146.41	184.53

Table 6: FIDs of methods with different priors (e.g., flow, EBM and Gaussian) in unsupervised image recovery.

5 Conclusion

Summary In this paper, we study modeling the latent space of data by the normalizing flow model and follow the philosophy of empirical Bayes to learn the latent space flow-based model (LFBM) from observed data. Specifically, we propose a novel top-down generative model with a latent space flow-based prior model. We propose a novel learning framework, in which the LFBM built on top of a top-down generative network serves as the prior model of the latent space, and is trained simultaneously with the top-down network in an MCMC-based MLE algorithm. We show that the learning algorithm with a short-run Langevin flow is a perturbation of MLE. The LFBM is more flexible and informa-
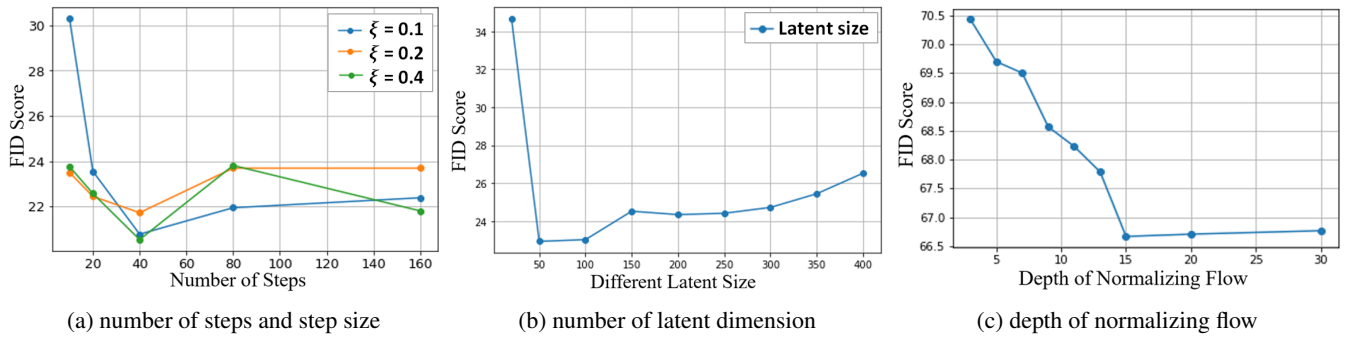
(a) number of steps and step size   (b) number of latent dimension   (c) depth of normalizing flow

Figure 3: Effects of different (a) numbers of Langevin steps and step sizes, (b) latent sizes, and (c) depths of normalizing flow.
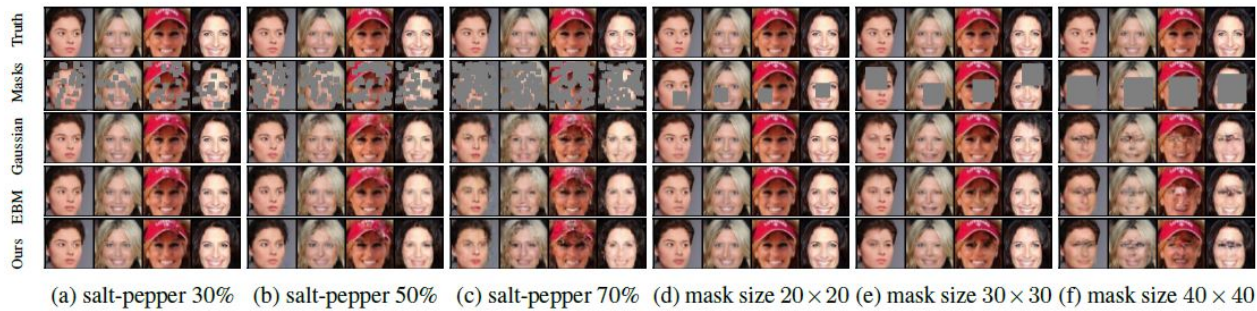

(a) salt-pepper 30%   (b) salt-pepper 50%   (c) salt-pepper 70%   (d) mask size 20×20 (e) mask size 30×30 (f) mask size 40×40

Figure 4: A comparison of unsupervised image recovery results by different methods on training images with different levels of occlusions. In each panel, the first row shows some original images that are used in the training process, the second row shows the corresponding occluded images with a certain occlusion level, and the third, fourth and fifth rows show the recovered images by models using Gaussian prior, EBM prior and normalizing flow prior, respectively.

(a) salt-pepper 30%   (b) salt-pepper 50%   (c) salt-pepper 70%   (d) mask size 20×20 (e) mask size 30×30 (f) mask size 40×40
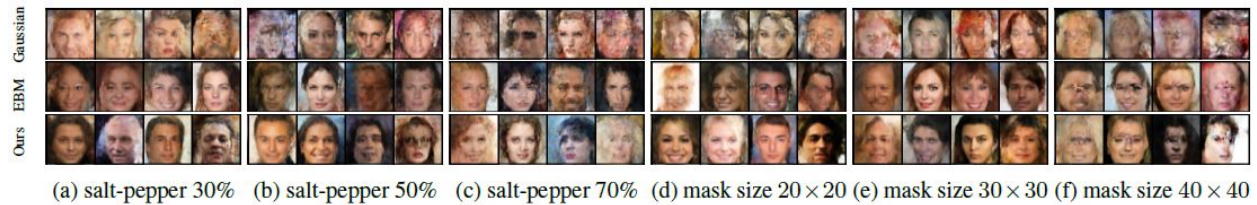
Figure 5: Image synthesis by models learned from incomplete images. Each panel represents a different level of occlusion.

tive than Gaussian prior and less biased to MLE than the un-normalized energy-based prior (LEBM). We test our LFBM on a variety of tasks to validate its effectiveness.

Limitation and Future Work In contrast to EBM prior, the normalizing flow prior allows for efficient and unbiased sampling and training of the prior distribution. However, these good properties do not come for free, and the usage of the flow-based prior implies the assumption that the latent variables are computationally efficient to normalize and can be generated by a finite sequence of invertible transformations. This assumption might potentially limit the expressive power of the prior distribution for the latent space. Given the fact that EBM prior is an unnormalized probability density and doesn't constrain the prior distribution by invertible functions, we may consider combining the representational flexibility of the EBM and the computational tractability of

the normalizing flow model to tackle the above limitations. To be specific, we specify a latent space EBM (LEBM) and a latent space flow-based model (LFBM) together to represent the prior distribution. We generalize the data space flow contrastive estimation (FCE) (Gao et al. 2020) to latent space for jointly training the LEBM and the LFBM without relying on MCMC. Flow contrastive estimation (Gao et al. 2020) can be considered as an improved version of noise contrastive estimation (NCE) (Gutmann and Hyvärinen 2010) for training EBM, where the original Gaussian noise is replaced by a normalizing flow to make it closer to the target distribution and a stronger contrast to the EBM. Meanwhile, the flow model is updated by approximately minimizing the Jensen-Shannon divergence between the flow model and the target distribution. In our future work, we will learn both LFBM and LEBM together in the latent space via FCE.

References

An, D.; Xie, J.; and Li, P. 2021. Learning Deep Latent Variable Models by Short-Run MCMC Inference With Optimal Transport Correction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 15415–15424.

Chen, X.; Kingma, D. P.; Salimans, T.; Duan, Y.; Dhariwal, P.; Schulman, J.; Sutskever, I.; and Abbeel, P. 2017. Variational Lossy Autoencoder. In *International Conference on Learning Representations (ICLR)*.

Ding, X.; and Gimpel, K. 2021. FlowPrior: Learning Expressive Priors for Latent Variable Sentence Models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 3242–3258.

Dinh, L.; Krueger, D.; and Bengio, Y. 2015. NICE: Non-linear Independent Components Estimation. In *International Conference on Learning Representations (ICLR) Workshop*.

Dinh, L.; Sohl-Dickstein, J.; and Bengio, S. 2016. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*.

Gao, R.; Nijkamp, E.; Kingma, D. P.; Xu, Z.; Dai, A. M.; and Wu, Y. N. 2020. Flow Contrastive Estimation of Energy-Based Models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7515–7525.

Gutmann, M.; and Hyvärinen, A. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 9, 297–304.

Han, T.; Lu, Y.; Zhu, S.; and Wu, Y. N. 2017. Alternating Back-Propagation for Generator Network. In *AAAI Conference on Artificial Intelligence (AAAI)*, 1976–1984.

Han, T.; Nijkamp, E.; Zhou, L.; Pang, B.; Zhu, S.-C.; and Wu, Y. N. 2020. Joint Training of Variational Auto-Encoder and Latent Energy-Based Model. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Annual Conference on Neural Information Processing Systems (NIPS)*, 6626–6637.

Huang, C.-W.; Touati, A.; Dinh, L.; Drozdzal, M.; Havaei, M.; Charlin, L.; and Courville, A. 2017. Learnable explicit density for continuous latent space and variational inference. *arXiv preprint arXiv:1710.02248*.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations (ICLR)*.

Kingma, D. P.; and Dhariwal, P. 2018. Glow: Generative Flow with Invertible 1x1 Convolutions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 10236–10245.

Kingma, D. P.; Salimans, T.; Jozefowicz, R.; Chen, X.; Sutskever, I.; and Welling, M. 2016. Improved variational inference with inverse autoregressive flow. *Annual Conference on Neural Information Processing Systems (NIPS)*, 29: 4743–4751.

Kingma, D. P.; and Welling, M. 2014. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations (ICLR)*.

Krizhevsky, A. 2009. Learning Multiple Layers of Features from Tiny Images. *Master's thesis, University of Toronto*.

Kumar, M.; Babaeizadeh, M.; Erhan, D.; Finn, C.; Levine, S.; Dinh, L.; and Kingma, D. 2020. VideoFlow: A Conditional Flow-Based Model for Stochastic Video Generation. In *International Conference on Learning Representations (ICLR)*.

Kumar, R.; Ozair, S.; Goyal, A.; Courville, A.; and Bengio, Y. 2019. Maximum entropy generators for energy-based models. *arXiv preprint arXiv:1901.08508*.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner., P. 1998. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, 2278–2324.

Liu, Z.; Luo, P.; Wang, X.; and Tang, X. 2015. Deep Learning Face Attributes in the Wild. In *IEEE International Conference on Computer Vision (ICCV)*, 3730–3738.

Ma, X.; Zhou, C.; and Hovy, E. H. 2019. MAE: Mutual Posterior-Divergence Regularization for Variational AutoEncoders. In *International Conference on Learning Representations (ICLR)*.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*.

Nijkamp, E.; Hill, M.; Zhu, S.; and Wu, Y. N. 2019. Learning Non-Convergent Non-Persistent Short-Run MCMC Toward Energy-Based Model. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 5233–5243.

Nijkamp, E.; Pang, B.; Han, T.; Zhou, L.; Zhu, S.; and Wu, Y. N. 2020. Learning Multi-layer Latent Variable Model via Variational Optimization of Short Run MCMC for Approximate Inference. In *European Conference on Computer Vision (ECCV)*, volume 12351, 361–378.

Pang, B.; Han, T.; Nijkamp, E.; Zhu, S.; and Wu, Y. N. 2020. Learning Latent Space Energy-Based Prior Model. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*.

Pang, B.; and Wu, Y. N. 2021. Latent Space Energy-Based Model of Symbol-Vector Coupling for Text Generation and Classification. In *International Conference on Machine Learning (ICML)*, volume 139, 8359–8370.

Pang, B.; Zhao, T.; Xie, X.; and Wu, Y. N. 2021. Trajectory Prediction With Latent Belief Energy-Based Model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 11814–11824.

Ping, W.; Peng, K.; Zhao, K.; and Song, Z. 2020. WaveFlow: A Compact Flow-based Model for Raw Audio. In *International Conference on Machine Learning (ICML)*, volume 119, 7706–7716.

Rezende, D. J.; and Mohamed, S. 2015. Variational Inference with Normalizing Flows. In *International Conference on Machine Learning (ICML)*, volume 37, 1530–1538.

Rubin, D. B.; and Thayer, D. T. 1982. EM algorithms for ML factor analysis. *Psychometrika*, 47(1): 69–76.

Shi, C.; Xu, M.; Zhu, Z.; Zhang, W.; Zhang, M.; and Tang, J. 2020. GraphAF: a Flow-based Autoregressive Model for Molecular Graph Generation. In *International Conference on Learning Representations (ICLR)*.

Xiao, Z.; Yan, Q.; and Amit, Y. 2019. Generative latent flow. *arXiv preprint arXiv:1905.10485*.

Xie, J.; Gao, R.; Zheng, Z.; Zhu, S.; and Wu, Y. N. 2019. Learning Dynamic Generator Model by Alternating Back-Propagation through Time. In *AAAI Conference on Artificial Intelligence (AAAI)*, 5498–5507.

Xie, J.; Gao, R.; Zheng, Z.; Zhu, S.; and Wu, Y. N. 2020. Motion-Based Generator Model: Unsupervised Disentanglement of Appearance, Trackable and Intrackable Motions in Dynamic Patterns. In *AAAI Conference on Artificial Intelligence (AAAI)*, 12442–12451.

Xie, J.; Lu, Y.; Zhu, S.; and Wu, Y. N. 2016. A Theory of Generative ConvNet. In *International Conference on Machine Learning (ICML)*, volume 48, 2635–2644.

Xie, J.; Zhu, Y.; Li, J.; and Li, P. 2022. A Tale of Two Flows: Cooperative Learning of Langevin Flow and Normalizing Flow Toward Energy-Based Model. In *International Conference on Learning Representations (ICLR)*.

Xing, X.; Gao, R.; Han, T.; Zhu, S.; and Wu, Y. N. 2022. Deformable Generator Networks: Unsupervised Disentanglement of Appearance and Geometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 44(3): 1162–1179.

Yang, G.; Huang, X.; Hao, Z.; Liu, M.-Y.; Belongie, S.; and Hariharan, B. 2019. Pointflow: 3d point cloud generation with continuous normalizing flows. In *International Conference on Computer Vision (ICCV)*, 4541–4550.

Zenati, H.; Foo, C. S.; Lecouat, B.; Manek, G.; and Chandrasekhar, V. R. 2018a. Efficient GAN-based anomaly detection. *arXiv preprint arXiv:1802.06222*.

Zenati, H.; Foo, C. S.; Lecouat, B.; Manek, G.; and Chandrasekhar, V. R. 2018b. Efficient GAN-based Anomaly Detection. *arXiv: 1802.06222*.

Zhang, J.; Xie, J.; and Barnes, N. 2020. Learning Noise-Aware Encoder-Decoder from Noisy Labels by Alternating Back-Propagation for Saliency Detection. In *European Conference on Computer Vision (ECCV)*, volume 12362, 349–366.

Zhang, J.; Xie, J.; Barnes, N.; and Li, P. 2021. Learning Generative Vision Transformer with Energy-Based Latent Space for Saliency Prediction. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*.

Zhu, Y.; Xie, J.; Liu, B.; and Elgammal, A. 2019. Learning Feature-to-Feature Translator by Alternating Back-Propagation for Generative Zero-Shot Learning. In *International Conference on Computer Vision (ICCV)*, 9843–9853.