Improving Sample Quality by Training and Sampling from Latent Energy

Zhisheng Xiao [*] [1] **Qing Yan** [*] [2] **Yali Amit** [2]

Abstract

In this paper, we present a general method that can improve the sample quality of pre-trained likelihood based generative models. Our method constructs an energy based model on the latent variable space that yields an energy function on samples produced by the pre-trained generative model. The energy based model is efficiently trained by maximizing the data likelihood, and after training, new samples in the latent space are generated from the energy based model and passed through the generator to produce samples in data space. We show that using our proposed method, we can greatly improve the sample quality of popular likelihood based generative models, such as normalizing flows and VAEs, with very little computational overhead.

1. Introduction

Recent advances in deep likelihood based generative models (Kingma & Welling, 2013; Rezende et al., 2014; Van den Oord et al., 2016; Salimans et al., 2017; Kingma & Dhariwal, 2018) enable the modeling of very high dimensional and complicated data such as natural images, sequences (Oord et al., 2016) and graphs (Kipf & Welling, 2016). Compared to Generative Adversarial Networks (GANs) (Goodfellow et al., 2014), these models can evaluate the likelihood of input data easily, return latent variables that are useful for downstream tasks while do not suffer from the instability and mode dropping issues (Salimans et al., 2016) of GAN training. However, GANs are still the state-of-the-art generative models for many generative tasks, because they can produce sharper and more realistic samples than non-adversarial likelihood-based models (Brock et al., 2018). Much effort has been devoted to improving the sample quality of likelihood based generative models (van den Oord et al., 2017; Behrmann et al., 2018; Huang et al., 2020; Dai & Wipf, 2019) by modifying the training objectives.

This paper aims at a slightly different task: we want to improve the sample quality of an existing generative model through a better sampling procedure. Note that in deep generative models, samples are usually obtained by sending latent variables through a deterministic transformation, where the latent variables are sampled from a pre-defined prior distribution. Controlling the temperature of sampling from the prior may produce better samples, but this comes at the cost of less diversity (Kingma & Dhariwal, 2018). Recently, there is a line of literature that improves the sample quality of pre-trained GANs (Tanaka, 2019; Che et al., 2020). They utilize the information contained in the discriminator of GANs to obtain better samples of the latent variable. In particular, (Che et al., 2020) sample latent variables from an energy-based model (EBM) defined jointly by the generator and discriminator.

Extension of these ideas to likelihood based models is non-trivial, because the discriminator of GANs is critical to provide guidance for moving the latent variables. We extend these methods to generative models without adversarial training by constructing a latent variable EBM that consists of the pre-trained generative model and an energy function. The EBM can be trained efficiently by maximizing the data likelihood, and we observe that training the EBM only adds a slight computational overhead, as the convergence is very fast. After convergence, new samples are produced by latent variables sampled from the EBM. We show that our method can effectively improve the sample quality of a variety of pre-trained generative models, including normalizing flows, VAEs and a combination of the two.

2. Background on Energy-Based Models

An Energy-based Model assumes a gibbs distribution

$$p_\theta(\mathbf{x}) = \frac{\exp(-E_\theta(\mathbf{x}))}{Z_\theta} \tag{1}$$

over data $\mathbf{x} \in \mathcal{X}$. $E_\theta(\mathbf{x})$ is the energy function with parameter θ, and $Z_\theta = \int_{\mathbf{x}} \exp(-E_\theta(\mathbf{x}))$ is the normalizing constant. When $\mathbf{x}$ is an image, $E_\theta(\mathbf{x})$ is usually chosen to be a convolutional neural network with scalar output (Du &

[*] Equal contribution [1] Computational and Applied Mathematics, The University of Chicago, Chicago, IL, USA [2] Department of Statistics, The University of Chicago, Chicago, IL, USA. Correspondence to: Zhisheng Xiao <zxiao@uchicago.edu>, Qing Yan <yanq@uchicago.edu>.

Second workshop on *Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models* (ICML 2020), Virtual Conference

Mordatch, 2019). The EBM can be trained by the maximum likelihood principle, namely minimizing the negative log likelihood $L(\theta) = \mathbb{E}_{\mathbf{x}\sim p_D}\left[-\log p_\theta(\mathbf{x})\right]$. This objective is known to have derivative

$$\frac{\partial L(\theta)}{\partial \theta} = \mathbb{E}_{p_D(\mathbf{x})}\left[\frac{\partial E_\theta\left(\mathbf{x}\right)}{\partial \theta}\right] - \mathbb{E}_{p_\theta(\mathbf{x}')}\left[\frac{\partial E_\theta\left(\mathbf{x}'\right)}{\partial \theta}\right] \quad (2)$$

where $\mathbf{x}'$ represents a sample drawn from the EBM. However, it is difficult to draw samples from such complex unnormalized distributions, and one typically needs to employ MCMC algorithms. One efficient MCMC algorithm in high dimensional continuous state spaces is Stochastic Gradient Langevin dynamics popularized in the statistics literature in the early 90's in (Amit et al., 1991) and introduced in the deep learning literature in (Welling & Teh, 2011). This algorithm initializes at $\mathbf{x}_0$ and runs updates

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \frac{\epsilon}{2}\nabla_{\mathbf{x}} E_\theta(\mathbf{x}) + \sqrt{\epsilon}\omega, \omega \sim \mathcal{N}(0, \mathbf{I}). \quad (3)$$

The continuous Langevin dynamics is guaranteed to produce samples from the target distribution. In practice we use a discrete approximation, which yields a Markov chain with invariant distribution close to the original target distribution.

3. Methodology

3.1. Exponential tilting of Generative Models

Suppose we have a pre-trained probabilistic generative model $p_{\phi^*}(\mathbf{x})$ over data space $\mathcal{X}$, we can define a new model by "exponential tilting" with energy function $E_\theta(\mathbf{x})$:

$$p_{\phi^*,\theta}(\mathbf{x}) = \frac{p_{\phi^*}(\mathbf{x})\exp\left(-E_\theta(\mathbf{x})\right)}{Z_{\phi^*,\theta}}, \quad (4)$$

where $Z_{\phi^*,\theta} = \int p_{\phi^*}(\mathbf{x})\exp\left(-E_\theta(\mathbf{x})\right)d\mathbf{x}$ is the corresponding normalizing constant. Since the plain EBM (1) is a special case with $p_{\phi^*} = \text{const}$, we can apply the training strategies in (2), (3) to $p_{\phi^*,\theta}(\mathbf{x})$. In particular, the Langevin dynamics to generate samples from $p_{\phi^*,\theta}(\mathbf{x})$ is

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \frac{\epsilon}{2}\nabla_{\mathbf{x}}\left(E_\theta(\mathbf{x}) - \log p_{\phi^*}(\mathbf{x})\right) + \sqrt{\epsilon}\omega.$$

Note that the derivative of the log likelihood of the original generative model $p_{\phi^*}(\mathbf{x})$ appears in the update, driving the Langevin dynamics to samples $\mathbf{x}$ with large likelihood and low energy simultaneously. However, this only works for generative models with tractable likelihood. More importantly, operating in the pixel space may be inefficient as it may require moving between different modes with barriers that cannot be easily overcome by the Langevin dynamics.

3.2. EBM in Latent Space

Many types of probabilistic generative models, including normalizing flows and VAEs, adopt a decoder structure in their generation process, namely there is a pre-defined prior distribution $p(\mathbf{z})$, and samples are generated by

$$\mathbf{z} \sim p(\mathbf{z}), \quad \mathbf{x} = G_{\phi^*}(\mathbf{z}).$$

We can therefore re-parametrize the EBM $p_{\phi^*,\theta}(\mathbf{x})$ in (4) with the latent variable $\mathbf{z}$ and obtain

$$p_{\phi^*,\theta}(\mathbf{z}) = \frac{p(\mathbf{z})\exp\left(-E_\theta(G_{\phi^*}(\mathbf{z}))\right)}{Z_{\phi^*,\theta}}. \quad (5)$$

When $p_{\phi^*,\theta}(\mathbf{z})$ is trained by maximizing the likelihood, the second term of (2) can also be re-parametrized to z space:

$$\mathbb{E}_{p_{\phi^*,\theta}(\mathbf{x})}\left[\frac{\partial E_\theta\left(\mathbf{x}\right)}{\partial \theta}\right] = \mathbb{E}_{p_{\phi^*,\theta}(z)}\left[\frac{\partial E_\theta(G_{\phi^*}(z))}{\partial \theta}\right] \quad (6)$$

See Appendix A for a simple derivation. Similarly, samples from $p_{\phi^*,\theta}(\mathbf{z})$ can be obtained through running the Langevin dynamics

$$\mathbf{z}_{i+1} = \mathbf{z}_i - \frac{\epsilon}{2}\nabla_{\mathbf{z}}\left(E_\theta(G_{\phi^*}(\mathbf{z})) - \log p(\mathbf{z})\right) + \sqrt{\epsilon}\omega. \quad (7)$$

Sometimes $\mathbf{z}$ has lower dimensionality than $\mathbf{x}$ and $p(\mathbf{z})$ can often be computed easily, therefore training and sampling from $p_{\phi^*,\theta}(\mathbf{z})$ is more efficient.

4. Related Work

Our work is closely related to recent literature that uses a discriminator to improve the sample quality of an existing GAN. Discriminator rejection sampling (Azadi et al., 2018) and Metropolis-Hastings GANs (Turner et al., 2018) use the discriminator as a criterion of accepting or rejecting samples from the generator. They are inefficient as many of samples may be rejected. Discriminator optimal transport (DOT) (Tanaka, 2019) and Discriminator Driven Latent Sampling (DDLS) (Che et al., 2020) both move the latent variable to make samples better fool the discriminator. In particular, (Tanaka, 2019) uses deterministic gradient descent in the latent space, while (Che et al., 2020) formulates an EBM on latent variables and use Langevin dynamics to sample latent variables. All these methods rely on the fact that the discriminator of a trained GAN is a good classifier on real/fake images, which is not applicable to likelihood based generative models.

When applied on VAEs, our method shares similarity with some recent literature that trains an auxiliary model to match the empirical latent distribution of an existing VAE, and samples are produced by latent variables generated by the auxiliary model. The auxiliary model can be another VAE (Dai & Wipf, 2019), normalizing flow (Xiao et al., 2019) or auto-regressive model (van den Oord et al., 2017). Our method use an EBM as the auxiliary model, but its purpose is to define the energy for generated samples. More importantly, other methods can improve the sample quality only when increasing the weight on the reconstruction term in the objective, which will make the latent representation less structured. In contrast, our method can improve the sample quality of VAE trained without modifying the objective.

Our method heavily relies on the progress of training deep EBMs. Similar to us, (Kumar et al., 2019) combine a generator and energy function together, but the generator and energy function are trained jointly in an adversarial game, which leads to instability issues. will Recently, (Du & Mordatch, 2019; Nijkamp et al., 2019a;b) successfully scales up the maximum likelihood learning of EBMs to high dimensional images. In particular, we follow (Nijkamp et al., 2019a;b) and use short run non convergent MCMC to train our EBMs.

5. Experiments

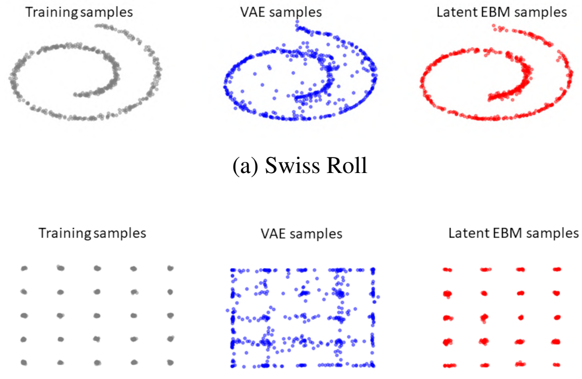
5.1. Toy dataset

To give a quick proof-of-concept, we apply our method on toy datasets (25-Gaussians and Swiss Roll) following the setting of (Tanaka, 2019). We first train a VAE on the training data, and then we fix the VAE and train a latent EBM as described in Section 3.2. The decoder and the energy function (which corresponds to the discriminator in GANs) have simple fully connected structure as described in (Tanaka, 2019). Note that we do not use normalizing flows on toy datasets, because vanilla flow is heavily constrained by the manifold structure of the prior distribution, making it very hard to model distributions like the 25-Gaussians.

We show qualitative results in Figure 1. We observe that although samples from VAEs can basically cover the shape of the true distribution, many samples still appear at low density regions. In contrast, by sampling and decoding latent variables obtained from the post-trained latent EBM, we can accurately preserve all modes in the target distribution while eliminating spurious modes in the 25-Gaussians case. In the Swiss Roll case, it is also clear that the EBM better captures the underlying data distribution.

5.2. Image dataset

In this section we evaluate the performance of the proposed latent EBM on MNIST, Fashion MNIST and CIFAR-10 dataset. We use different decoder based generative models $p_{\phi^*}(\mathbf{x})$, including normalizing flow, VAE and GLF, which uses a latent flow model (Xiao et al., 2019) that combines a deterministic auto-encoder and a normalizing flow on the latent variables. Note that our main focus is on the relative improvements of sampling from the EBMs over sampling from base generative models, and therefore the performances of the base generative models may not be state-of-the-art. In fact, we adopt relatively simple network structures for convenience.



(a) Swiss Roll

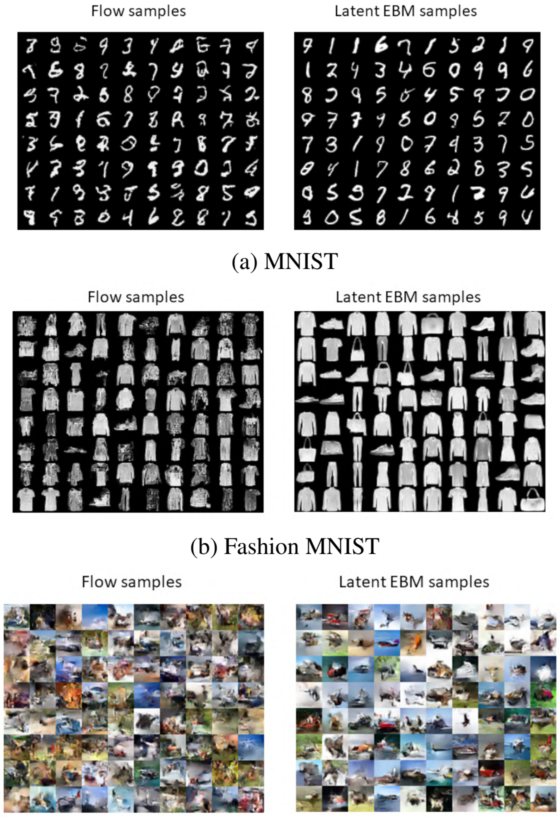
(b) 25-Gaussians

Figure 1. Applying latent EBM to VAEs trained on Swiss Roll and 25-Gaussians datset.

As in (Du & Mordatch, 2019), we use a convolutional network with scalar outputs as E_θ. We adopt short-run non-persistent MCMC, so the Langevin dynamics (7) is run with $\mathbf{z}_0$ initialized from $p(\mathbf{z})$ for a small number of steps. We fix $G_{\phi^*}(\mathbf{z})$ and E_θ is trained by maximum likelihood. For details on the settings of our experiments, see Appendix B. It should be noted that the energy function only introduces a small parameter overhead. For example, its number of parameters is less than 5% of that of GLOW.

We show some qualitative results of training our proposed latent EBMs on top of a GLOW (Kingma & Dhariwal, 2018) model in Figure 2. From Figure 2, we clearly observe that samples generated by latent variables obtained from the latent EBMs have higher quality than samples from the base generative model (i.e., decoding latent variables from prior distribution). On MNIST and Fashion MNIST, samples obtained through the latent EBM have smoother shapes than samples from the GLOW. On CIFAR-10, the latent EBM effectively corrects the noisy backgrounds of the samples generated by the GLOW. We illustrate the process of Langevin dynamics sampling from the latent EBM in Figure 3, where we generate samples for every 10 iterations. Apparently the the Langevin dynamics is going towards latent variables that produce more semantically meaningful and sharp samples.

More qualitative examples, including results of training latent EBMs on top of VAE and GLF are presented in Appendix C. In Figure 5, we observe that the VAE+latent EBM generates sharper samples than VAE alone. However, it should be noted that, since our EBM operates on the latent space, the overall sample quality is constrained by the capacity of the base generative models.
Our observation on the improvements of sample quality can be confirmed by quantitative results in Table 1, where we compare the FID scores (Heusel et al., 2017) of different



(a) MNIST

(b) Fashion MNIST

(c) CIFAR-10

Figure 2. Applying latent EBM to GLOW trained on MNIST, Fashion and CIFAR-10. **Left**: samples generated by $\mathbf{z}$'s from the prior. **Right**: samples generated by $\mathbf{z}$'s from $p_{\phi^*,\theta}(\mathbf{z})$.

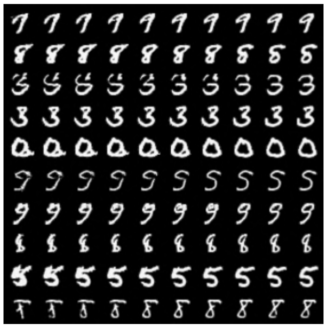


Figure 3. MNIST Langevin dynamics visualization, initialized at samples from prior (the leftmost column).

	MNIST	Fashion	CIFAR-10
GLOW	29.4	58.7	76.2
GLOW + EBM	12.3	41.6	67.8
VAE	18.9	57.1	139.6
Two-stage VAE	19.3	55.7	134.6
VAE + flow prior	18.6	52.3	128.2
VAE + EBM	16.0	38.1	108.4
GLF	14.2	32.5	96.6
GLF + EBM	12.1	25.3	85.1
EBM on data	25.5	39.3	80.6

Table 1. Comparing the FID scores of base generative models and generative models + exponential tilting with latent EBMs. Scores are computed using 10000 generated samples and real samples from the test set.

models. We see that sampling latent variables from the latent EBM significantly improves the quality of generated samples over directly sampling from $p_\theta(\mathbf{x})$. In addition, we see that methods mentioned in Section 4 do not improve the sample quality of VAEs *without posing a large weight on reconstruction term*, while our method can generate better samples without changing the VAE's objective, leading to good sample quality *and* structured latent representation. For completeness, we also present results of training EBMs directly on pixel space using the same model structures. The results are in the same range as the latent EBM models, but we observe that training EBMs on data is more sensitive to hyper-parameter settings and more computationally expensive.

5.3. Overfitting issue of training latent EBMs

As pointed out in (Grathwohl et al., 2019; Nijkamp et al., 2019a), instability and overfitting are frequently observed when training Energy-based Models. Overfitting happens when the energy of training samples are much lower than samples drawn from the EBM, which causes the model to produce worse samples. We find heuristic approaches such as energy regularization and gradient clipping not helpful in preventing overfitting, so we simply stop the training of the latent EBM when sample quality deteriorates. We believe that future studies in improving the stability of EBM training can further boost the performance of our method.

5.4. Training time

One training step of EBM requires obtaining a sample by running multiple steps of MCMC, and therefore it is much slower than one training step of the base generative model. However, we find that very few training iterations are needed to achieve the results in Table 1. Specifically, we only train latent EBMs for 200 steps when $G_{\phi^*}(\mathbf{z})$ is a GLOW or GLF, and 1000 steps when $G_{\phi^*}(\mathbf{z})$ is a VAE. Here a step refers to **one** batch, not an entire epoch. These numbers are several orders of magnitude smaller than the training steps needed for the base generative models. Therefore, our method does not add much computational overhead. As a comparison, training EBMs on pixel space typically requires more than 100k steps.

6. Conclusion

In this paper, we propose to train an Energy-based model on the latent space of pre-trained generative models. We show that with little computational overhead, we can improve the sample quality of a variety of generative models, including normalizing flow and VAE, by sampling latent variables from the EBM. Our method also provides a general framework that connects Energy-based models and other likelihood based generative models. We believe this connection is an interesting direction for future research.

References

Amit, Y., Grenander, U., and Piccioni, M. Structural image restoration through deformable template. *Journal of the American Statistical Association*, 86(414):376–387, 1991.

Azadi, S., Olsson, C., Darrell, T., Goodfellow, I., and Odena, A. Discriminator rejection sampling, 2018.

Behrmann, J., Grathwohl, W., Chen, R. T., Duvenaud, D., and Jacobsen, J.-H. Invertible residual networks. *arXiv preprint arXiv:1811.00995*, 2018.

Brock, A., Donahue, J., and Simonyan, K. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Che, T., Zhang, R., Sohl-Dickstein, J., Larochelle, H., Paull, L., Cao, Y., and Bengio, Y. Your gan is secretly an energy-based model and you should use discriminator driven latent sampling. *arXiv preprint arXiv:2003.06060*, 2020.

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

Du, Y. and Mordatch, I. Implicit generation and generalization in energy-based models. *arXiv preprint arXiv:1903.08689*, 2019.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks. arxiv 2014. *arXiv preprint arXiv:1406.2661*, 2014.

Grathwohl, W., Wang, K.-C., Jacobsen, J.-H., Duvenaud, D., Norouzi, M., and Swersky, K. Your classifier is secretly an energy based model and you should treat it like one. *arXiv preprint arXiv:1912.03263*, 2019.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pp. 6626–6637, 2017.

Huang, C.-W., Dinh, L., and Courville, A. Augmented normalizing flows: Bridging the gap between generative flows and latent variable models. *arXiv preprint arXiv:2002.07101*, 2020.

Kingma, D. P. and Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pp. 10215–10224, 2018.

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

Kipf, T. N. and Welling, M. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

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

Nalisnick, E., Matsukawa, A., Teh, Y. W., Gorur, D., and Lakshminarayanan, B. Do deep generative models know what they don't know? *arXiv preprint arXiv:1810.09136*, 2018.

Nijkamp, E., Hill, M., Han, T., Zhu, S.-C., and Wu, Y. N. On the anatomy of mcmc-based maximum likelihood learning of energy-based models. *arXiv preprint arXiv:1903.12370*, 2019a.

Nijkamp, E., Hill, M., Zhu, S.-C., and Wu, Y. N. Learning non-convergent non-persistent short-run mcmc toward energy-based model. In *Advances in Neural Information Processing Systems*, pp. 5233–5243, 2019b.

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

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

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training gans. In *Advances in neural information processing systems*, pp. 2234–2242, 2016.

Salimans, T., Karpathy, A., Chen, X., and Kingma, D. P. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017.

Tanaka, A. Discriminator optimal transport. In *Advances in Neural Information Processing Systems*, pp. 6813–6823, 2019.

Turner, R., Hung, J., Frank, E., Saatci, Y., and Yosinski, J. Metropolis-hastings generative adversarial networks. *arXiv preprint arXiv:1811.11357*, 2018.

Van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al. Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems*, pp. 4790–4798, 2016.

van den Oord, A., Vinyals, O., et al. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, pp. 6306–6315, 2017.

Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pp. 681–688, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

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

A. Proof of (6)

Since p_{ϕ^*} is generated through a deterministic mapping G_{ϕ^*} from the latent space $\mathcal{Z}$ to the observation space $\mathcal{X}$, for any function f on $\mathcal{X}$ we have:

$$\begin{aligned}\mathbb{E}_{p_{\phi^*}}[f(\mathbf{x})] &= \int_{\mathcal{X}} f(\mathbf{x}) p_{\phi^*}(\mathbf{x}) d\mathbf{x} \\ &= \int_{\mathcal{Z}} f(G_{\phi^*}(\mathbf{z})) p(\mathbf{z}) d\mathbf{z} = \mathbb{E}_{p(\mathbf{z})}[f(G_{\phi^*}(\mathbf{z})].\end{aligned}$$

Therefore

$$Z_{\phi^*,\theta} = \mathbb{E}_{p_{\phi^*}(\mathbf{x})}\left[e^{-E_\theta(\mathbf{x})}\right] = \mathbb{E}_{p(\mathbf{z})}\left[e^{-E_\theta(G_{\phi^*}(\mathbf{z}))}\right]$$

Take derivative w.r.t θ we can get

$$\begin{aligned}\frac{\partial \log Z_{\phi^*,\theta}}{\partial \theta} &= -\mathbb{E}_{p_{\phi^*,\theta}(\mathbf{x})}\left[\frac{\partial E_\theta(\mathbf{x})}{\partial \theta}\right] \\ &= -\mathbb{E}_{p(\mathbf{z})}\left[\frac{e^{-E_\theta(G_{\phi^*}(\mathbf{z}))}}{Z_{\phi^*,\theta}} \frac{\partial E_\theta(G_{\phi^*}(\mathbf{z}))}{\partial \theta}\right] \\ &= -\mathbb{E}_{p_{\phi^*,\theta}(\mathbf{z})}\left[\frac{\partial E_\theta(G_{\phi^*}(\mathbf{z}))}{\partial \theta}\right],\end{aligned}$$

which is exactly (6).

B. Experimental Settings

B.1. Base generative models

We first introduce training settings of the base generative models that we used in our experiments. We train GLOW basically following the settings provided in (Nalisnick et al., 2018). For MNIST and Fashion MNIST,we use a GLOW architecture of 2 blocks of 16 affine coupling layers, squeezing the spatial dimension in between the 2 blocks. For the coupling function, we use a 3-layer Highway network with 64 hidden channels. For CIFAR-10, we use 3 blocks of 32 affine coupling blocks, applying the multi-scale architecture between each block. The coupling function is a 3-layer Highway network with 256 hidden channels. Note that we modify the model size to fit in a single GPU for training. For MNIST and Fashion MNIST, we train the GLOW for 128 epochs with batch size 64 and Adam optimizer with fixed learning rate 5×10^{-4}. For CIFAR-10, we train the GLOW for 256 epochs with batch size 64 and Adam optimizer with fixed learning rate 5×10^{-4}.

Our use the DCGAN (Radford et al., 2015) structure on the decoders of our VAEs, and the encoders are designed to be symmetric to the decoder. We use latent dimension 100 for all experiments. For MNIST and Fashion datasets, we use binary cross entropy as reconstruction loss, while for CIFAR-10, we use MSE loss. All VAEs are trained for 256 epochs with batch size 128 and Adam optimizer with fixed learning rate 1×10^{-3}.

For GLF adopt the same encoder-decoder structure as in our settings for training VAEs. We use latent dimension 64 for all experiments. The normalizing flow for matching the latent distribution is a simple GLOW network with 4 affine coupling layers, each consists of one fully connected layer with 256 units. The AE and the flow are jointly trained for 256 epochs with batch size 128 and Adam optimizer with fixed learning rate 1×10^{-3}.

B.2. Energy based models

We used a simplified version of the network structure described in (Du & Mordatch, 2019) to define our E_θ. In particular, our method consists of 3 resnet blocks with 64 hidden channels and 3 resent blocks with 128 hidden channels, followed by Global Sum Pooling and a FC layer. We also find the network structure in (Nalisnick et al., 2018), which has much less parameters, leads to only slightly worse performances. Therefore, their energy function can be used for parameter efficiency.

Unlike (Du & Mordatch, 2019; Nijkamp et al., 2019a) where the Langevin dynamics is dominated by the gradient, we find our latent EBMs work well with balanced noise and gradient in (7). For Langevin dynamics, we use $\epsilon = 0.01$ and run the chain for 60 steps. We find adding a small amount (with coefficient 0.1) of energy regularization is helpful for avoiding over-fitting early in the training. After training, we find sampling latent variables with longer chain leads to better performances. We generate samples from $p_{\phi^*,\theta}(\mathbf{z})$ by running the chain for 100 steps.

For EBMs on the pixel space, we find short-run non-persistent training as described in (Nijkamp et al., 2019a) hard to converge on MNIST and Fashion MNIST, so we follow the setting in (Du & Mordatch, 2019), where they use persistent initialization for the Langevin dynamics. They maintain a sample replay buffer during the training, and samples from the buffer are used to initialize the chain. We follow the hyper-parameter settings in (Du & Mordatch, 2019),and we train EBMs on MNIST and Fashion MNIST for 20k steps, and on CIFAR-10 for 100k steps. Note that we train less number of steps on CIFAR-10 than the open source implementation of (Du & Mordatch, 2019), because it takes prohibitively long time due to the hardware constraint. We find our samples qualitatively comparable to those of (Du & Mordatch, 2019) (see Figure 7), but we are unable to match their reported FID scores on CIFAR-10, possibly due to not training long enough. After training, new samples are generated from chains initialized from the replay buffer.

C. Additional Qualitative Results

In this section, we show some additional qualitative results. In Figure 4, we presents more examples of samples from GLOW and GLOW + latent EBM, in addition to Figure 2 in the main text. In Figure 5 we show samples from VAE and VAE + latent EBM. In Figure 6, we show samples from GLF and GLF + latent EBM. In all of these experiments, we clearly observe that latent EBMs improve the sample quality of base generative models. Finally, in Figure 7, we show samples from EBMs trained on pixel space.

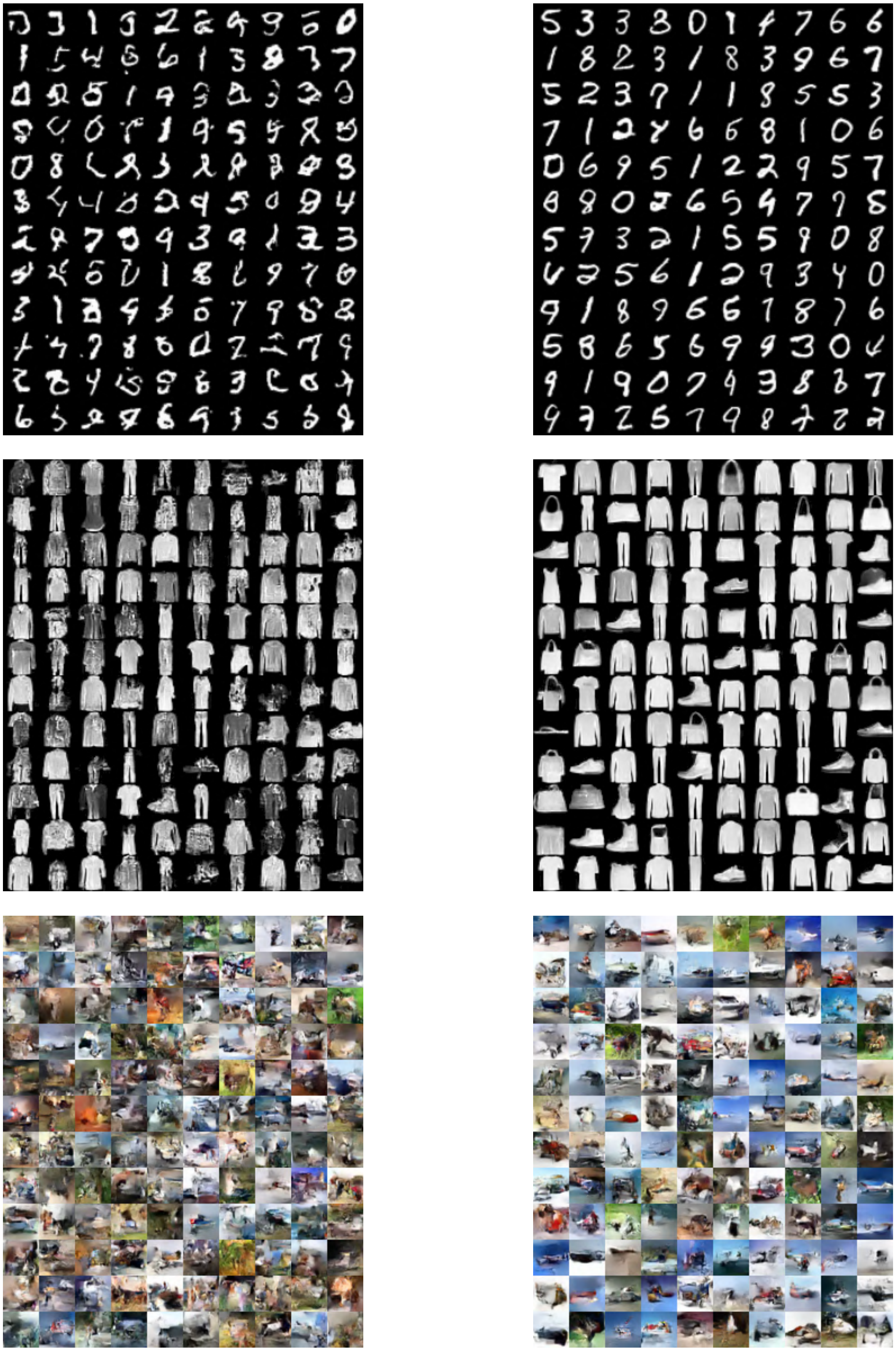


Figure 4. Additional qualitative results of GLOW + atent EBM on MNIST, Fashion and CIFAR-10. **Left**: samples generated by $\mathbf{z}$'s from the prior. **Right**: samples generated by $\mathbf{z}$'s from $p_{\phi^*,\theta}(\mathbf{z})$.

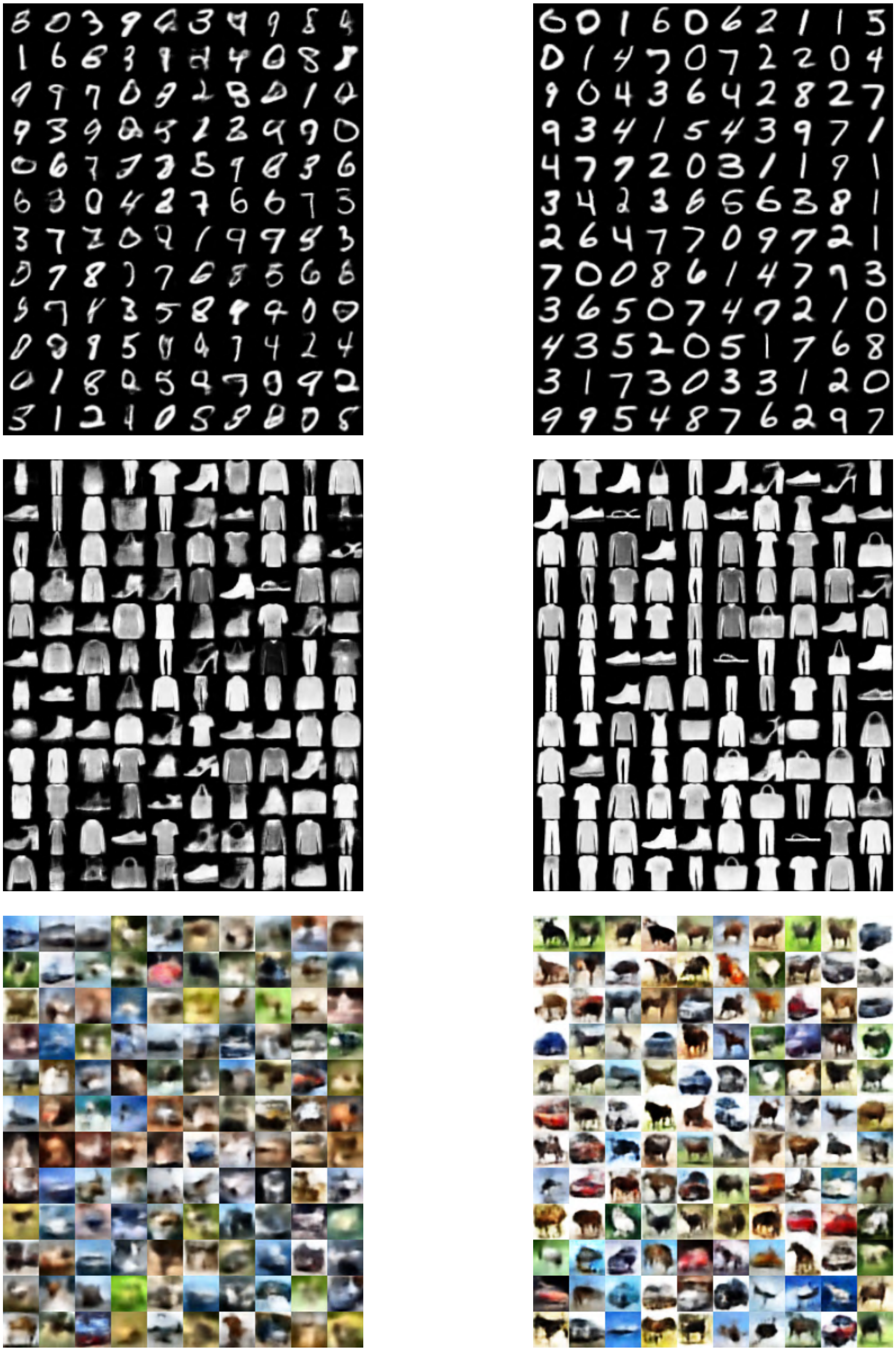


Figure 5. Qualitative results of VAE + latent EBM on MNIST, Fashion and CIFAR-10. **Left**: samples generated by $\mathbf{z}$'s from the prior. **Right**: samples generated by $\mathbf{z}$'s from $p_{\phi^*,\theta}(\mathbf{z})$.

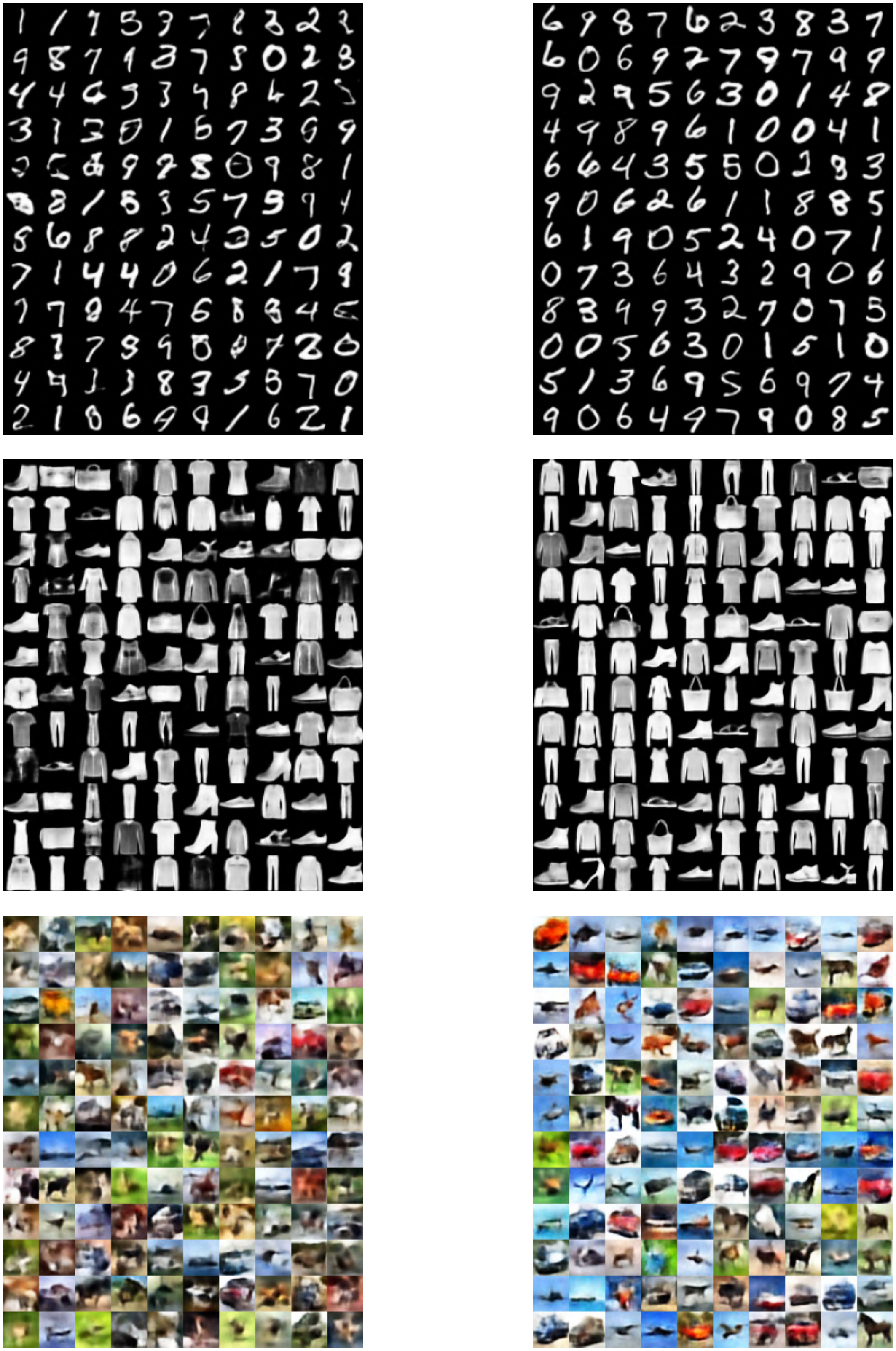
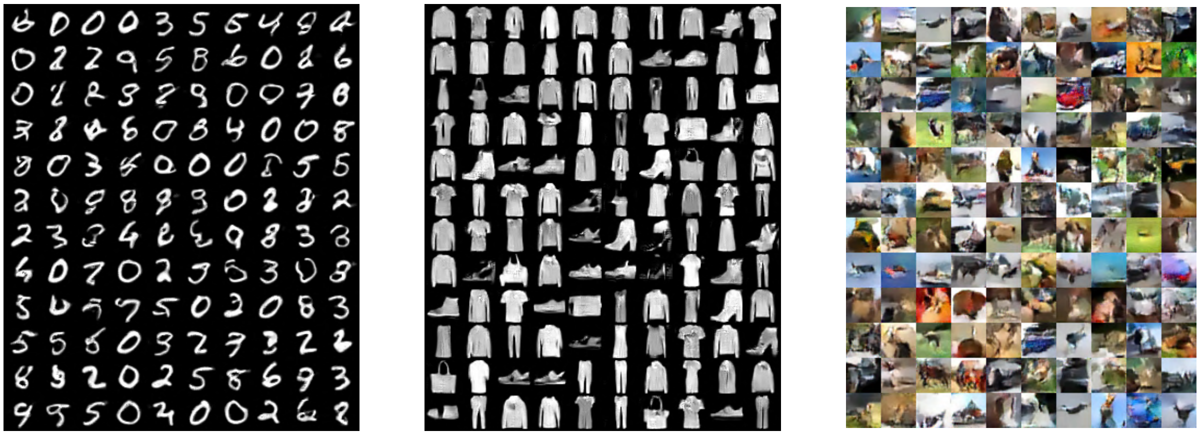


Figure 6. Qualitative results of GLF + latent EBM on MNIST, Fashion and CIFAR-10. **Left**: samples generated by $\mathbf{z}$'s from the prior. **Right**: samples generated by $\mathbf{z}$'s from $p_{\phi^*,\theta}(\mathbf{z})$.



(a) MNIST

(b) Fashion MNIST

(c) CIFAR-10

Figure 7. Qualitative samples from EBMs trained on pixel space.