On Learning Non-Convergent Non-Persistent Short-Run MCMC Toward Energy-Based Model.

Reply to Reviewer 2: Thank you for the insightful and comprehensive summary of our work.

Q1: About training time. A1: As you have pointed out, each iteration requires computing K derivatives of the CNN. As an example, $100,000$ model parameter updates for 64×64 CelebA with $K = 100$ and $n_f = 64$ on 4 Titan Xp GPUs take 16 hours. We will add such information in revision.
Q2: Dynamic K. A2: Following your advice, we conducted experiments with random $K \in [100, 120]$ for training. We can still learn short-run MCMC successfully. This corresponds to residual network with random number of layers.
Q3: Using different K for training and sampling. A3: Following your advice, we conducted an experiment on CelebA, where we train the model with K_1 and test the trained model by running MCMC with K_2 steps. The Figures below depict training with $K_1 \in \{100, 50\}$ and varied K_2 for sampling. Note, over-saturation occurs for $K_2 > K_1$.

Transition with $K_1 = 100$ for training and varying K_2 for sampling.

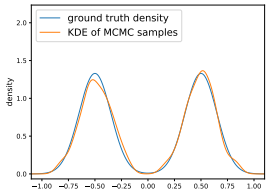
$K_2 = 0$ 20 40 60 80 100 120 140 160 180 200

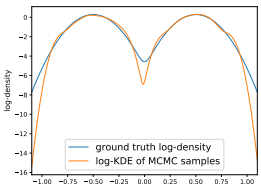
Transition with $K_1 = 50$ for training.

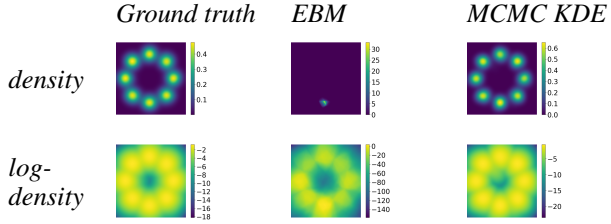
$K_2 = 25$ 50 100

Reply to Reviewer 3: Thank you for the insightful comments.

Q1: About theoretical justification. A1: Your comment is well taken, and we shall continue to try our best on theoretical understanding. About the generalized moment matching estimator, classical statistical theory shows that the estimator is asymptotically unbiased, and the variance of the estimator can also be derived (see Supplementary 2.2). The convergence of the learning algorithm follows Robbins-Monro stochastic approximation because $q_\theta(x)$ can be sampled exactly.
Q2: About computing log-likelihood. A2: Following your advice, we computed log-likelihood. The prior model is $z \sim p_0(z)$ which is the uniform distribution. After training, we learn the dynamics $x = M_\theta(z)$, where M_θ consists of K-steps of gradient descent dynamics (the noise term of the Langevin dynamics is negligible compared to the gradient term after learning). Under this flow dynamics $p(x) = p_0(z)/\det(\partial M_\theta(z)/\partial z))$ (where we used GELU, a differentiable version of ReLU). For each observed x, we can obtain z as described in Section 3.4. Then we compute the Jacobian and its determinant as the product of the eigenvalues of the Jacobian. In our preliminary results, the log-likelihood computation is feasible for images of size $32 \times 32 \times 3$. For a small batch of 16 images, we have obtained a rough, preliminary estimate average 4.12 number of bits per data dimension. We will include log-likelihood results in revision. We shall also try to implement Burda et al. (thanks for the reference).
Q3: 1D and 2D toy examples. A3: Following your advice, we did 1D and 2D experiments. We plot the density and log-density of the true model, the learned EBM, and the kernel density estimate (KDE, like histogram) of the MCMC samples. The density of the MCMC samples matches the true density closely. The learned energy captures the modes of the true density, but is of a much bigger scale, so that the learned EBM density is of much lower entropy or temperature (so that the density focuses on the global energy minimum). This is consistent with our theoretical understanding.







Reply to Reviewer 4: Thank you for the interesting questions.

Q1: About ability to reconstruct. A1: You are right. It is due to short-run non-mixing, i.e., different starting point z leads to different x after K-step MCMC, and K is fixed, so that a mapping is well-defined $z = M_\theta(x)$, where M_θ consists of K-steps of gradient descent dynamics (the noise term of the Langevin dynamics is negligible compared to the gradient term after learning). M_θ can be considered a flow model.
Q2: Large K is favored. A2: We believe larger K achieves improved synthesis results because larger K leads to a better learned flow model $M_\theta(z)$ (as a residual network with more layers). Our experience suggests that increasing K much further beyond 100 does not lead to much improved results.
Q3: Monte Carlo EM. A3: Our short-run MCMC always initializes from noise images sampled from the uniform distribution. In Monte Carlo EM, one may initialize K-step MCMC that samples from the posterior distribution of the latent variables from the currently generated samples, i.e., running the so-called persistent chains with warm start. Of course, one may also initialize from the same noise distribution, i.e., cold start. In that case, the short-run MCMC may be interpreted as a variational inference model. We have used this method to learn a latent EBM similar to Boltzmann machine but with continuous latent variables, where we use short-run MCMC for both inference and synthesis.