NEURAL INFORMATION PROCESSING SYSTEMS

Learning Energy-Based Prior Model with Diffusion-Amortized MCMC

Peiyu Yu[1], Yaxuan Zhu[1], Sirui Xie[1], Xiaojian Ma[1,2], Ruiqi Gao[3], Ying Nian Wu[1], Song-Chun Zhu[2]

[1]UCLA, [2]BIGAI, [3]Google DeepMind

Overview

How to learn the latent space energy-based prior model through MLE?

MLE Pros:

- Principled
- Good asymptotic properties
- Etc.

MLE Cons:

- Requires MCMC
- Short-run MCMC is biased
- Long-run MCMC is slow

Our Solution:

Diffusion-Based Amortization of MCMC

Method

Background: Latent Space Energy-Based Prior Model

Complete data distribution:

$$p_{\boldsymbol{\theta}}(\boldsymbol{z},\boldsymbol{x}) := p_{\boldsymbol{\alpha}}(\boldsymbol{z})p_{\boldsymbol{\beta}}(\boldsymbol{x}|\boldsymbol{z}),\ p_{\boldsymbol{\alpha}}(\boldsymbol{z}) := \frac{1}{Z_{\boldsymbol{\alpha}}}\exp\left(f_{\boldsymbol{\alpha}}(\boldsymbol{z})\right)p_0(\boldsymbol{z}),$$

Learning gradients of the prior model:

$$\delta_{\boldsymbol{\alpha}}(\boldsymbol{x}) := \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}\left[\nabla_{\boldsymbol{\alpha}} f_{\boldsymbol{\alpha}}(\boldsymbol{z})\right] - \mathbb{E}_{p_{\boldsymbol{\alpha}}(\boldsymbol{z})}\left[\nabla_{\boldsymbol{\alpha}} f_{\boldsymbol{\alpha}}(\boldsymbol{z})\right],\ \delta_{\boldsymbol{\beta}}(\boldsymbol{x}) := \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})}\left[\nabla_{\boldsymbol{\beta}} \log p_{\boldsymbol{\beta}}(\boldsymbol{x}|\boldsymbol{z})\right].$$

Langevin dynamics for sampling:

$$\boldsymbol{z}_{t+1} = \boldsymbol{z}_t + \frac{s^2}{2}\nabla_{\boldsymbol{z}_t}\log\pi(\boldsymbol{z}_t) + s\boldsymbol{w}_t,\ t = 0, 1, ..., T-1,\ \boldsymbol{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$$

Method

Amortizing MCMC

$$q_{\phi_k} \leftarrow \arg\min_{q_\phi \in \mathcal{Q}} \mathcal{D}[q_{\phi_{k-1},T} || q_\phi], \; q_{\phi_{k-1},T} := \mathcal{K}_T q_{\phi_{k-1}}, \; q_{\phi_0} \approx \pi_0, \; k = 0, ..., K-1.$$

Given the transition kernel K:

a) Employ a T-step short-run LD initialized with the current sampler $q_{\boldsymbol{\phi}_{k-1}}$ to approximate $K_T q_{\boldsymbol{\phi}_{k-1}}$ as the target distribution of the current sampler

b) Update the current sampler $q_{\boldsymbol{\phi}_{k-1}}$ to $q_{\boldsymbol{\phi}_k}$

Method

Diffusion-based amortization:

$$\phi_{k-1}^{(i+1)} \leftarrow \phi_{k-1}^{(i)} - \eta \nabla_{\phi} \mathbb{E}_{\epsilon,\lambda}\left[\|\epsilon(z_\lambda) - \epsilon\|_2^2\right],\ \phi_k^{(0)} \leftarrow \phi_{k-1}^{(M)},\ i = 0, 1, ..., M-1$$

a) For the choice of q_{ϕ}, let us consider distilling the gradient field of target q in each iteration, so that the resulting sampler is close to the target distribution.

b) This naturally points to the DDPMs. To be specific, learning a DDPM with ϵ-prediction parameterization is equivalent to fitting the finite-time marginal of a sampling chain resembling annealed LD.

c) We can plug in the objective of DDPM, which is a lower bound of logq_{ϕ} , to obtain the gradient-based update rule for q_{ϕ}.

Method

Diffusion-based amortization:

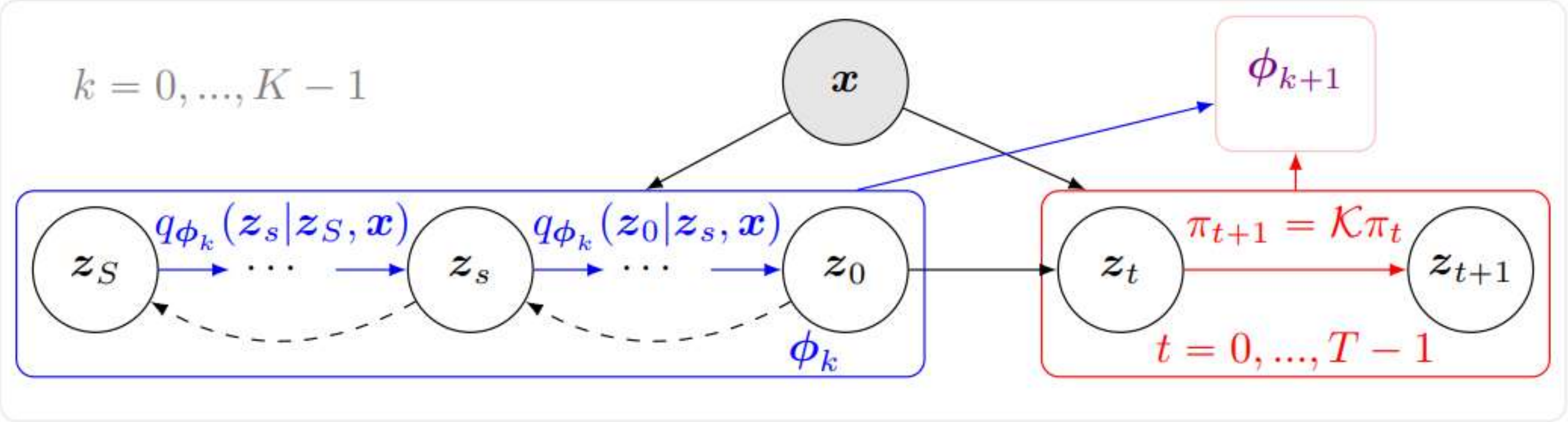
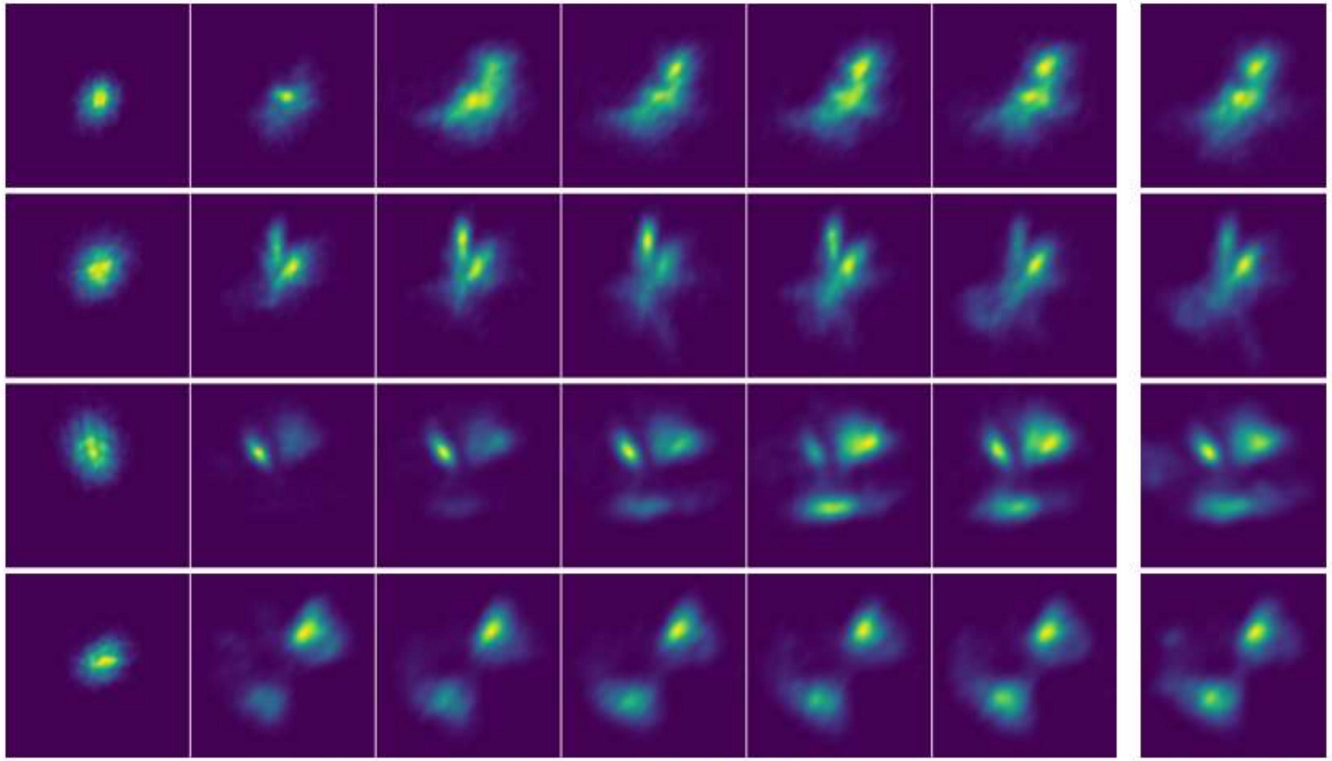


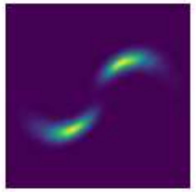
Figure 1: **Learning the DAMC sampler.** The training samples for updating the sampler to ϕ_{k+1} is obtained by T-step short-run LD, initialized with the samples from the current learned sampler ϕ_k. Best viewed in color.

Experiments

Amortizing Long-Run (1k-3k stps.) MCMC



(a) Evolution of the learned posterior distributions (b) GT



The 2-arm pinwheel-shaped prior distribution used in the toy example.

Neural likelihood experiment:

a) The true posterior distributions are multimodal.
b) Posterior obtained by performing LD sampling until convergence.
c) Our method can amortize long-run chains w/ the length of 1k-3k steps.

Experiments

Generation and Inference: Prior and Posterior Sampling

(a) SVHN (b) CelebA (c) CIFAR-10 (d) CelebA-HQ

Figure 2: **Samples generated from the DAMC sampler and LEBM** trained on SVHN, CelebA, CIFAR-10 and CelebA-HQ datasets. In each sub-figure, the first four rows are generated by the DAMC sampler. The last four rows are generated by LEBM trained with the DAMC sampler.

Model	SVHN		CelebA		CIFAR-10		CelebA-HQ	
	MSE	FID	MSE	FID	MSE	FID	MSE	FID
VAE [1]	0.019	46.78	0.021	65.75	0.057	106.37	0.031	180.49
2s-VAE [48]	0.019	42.81	0.021	44.40	0.056	72.90	-	-
RAE [49]	0.014	40.02	0.018	40.95	0.027	74.16	-	-
NCP-VAE [50]	0.020	33.23	0.021	42.07	0.054	78.06	-	-
Adaptive CE* [41]	0.004	26.19	0.009	35.38	**0.008**	65.01	-	-
ABP [51]	-	49.71	-	51.50	0.018	90.30	0.025	160.21
SRI [24]	0.018	44.86	0.020	61.03	-	-	-	-
SRI (L=5) [24]	0.011	35.32	0.015	47.95	-	-	-	-
LEBM [22]	0.008	29.44	0.013	37.87	0.020	70.15	0.025	133.07
Ours-LEBM	**0.002**	21.17	**0.005**	35.67	0.015	60.89	**0.023**	89.54
Ours-DAMC		**18.76**		**30.83**		**57.72**		**85.88**

- Prior model learned by our method demonstrates better generation quality.
- Posterior samples from the proposed method produces sharper reconstruction results.

NEURAL INFORMATION PROCESSING SYSTEMS

Thank you