# Learning Energy-based Model via Dual-MCMC Teaching

Jiali Cui    Tian Han
Stevens Institute of Technology

NEURAL INFORMATION PROCESSING SYSTEMS

## Preliminary I: Energy-based Model

**Energy-based Model (EBM).** The EBM can be defined with probability density as

$$\pi_\alpha(\mathbf{x}) = \frac{1}{Z(\alpha)} \exp\left[f_\alpha(\mathbf{x})\right]$$

**Maximum Likelihood Estimation (MLE).** With observed examples $\{\mathbf{x}^{(i)}, i = 1, 2, ..., n\}$, learning of EBM can be done via MLE with the gradient computed as

$$\frac{\partial}{\partial \alpha} L_\pi(\alpha) = \mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[\frac{\partial}{\partial \alpha} f_\alpha(\mathbf{x})\right] - \mathbb{E}_{\pi_\alpha(\mathbf{x})}\left[\frac{\partial}{\partial \alpha} f_\alpha(\mathbf{x})\right]$$

**Limitation.** The MLE learning requires sampling from the EBM, which can be achieved by Markov Chain Monte Carlo (MCMC) sampling, such as the Langevin dynamics,

$$\mathbf{x}_{\tau+1} = \mathbf{x}_\tau + s\frac{\partial}{\partial \mathbf{x}_\tau} \log \pi_\alpha(\mathbf{x}_\tau) + \sqrt{2s}U_\tau$$

**However**, such Langevin dynamics typically start from noise points, which may take a long time to converge and mix between different modes.

## Preliminary II: Generator Model

**Generator Model.** The generator model seeks to explain the observation signal $\mathbf{x}$ by a latent vector $\mathbf{z}$ and can be specified as

$$p_\theta(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z})$$

where $p(\mathbf{z})$ is the known prior distribution such as unit Gaussian, and $p_\theta(\mathbf{x}|\mathbf{z})$ is the generation model.

**Maximum Likelihood Esitmation (MLE).** The MLE learning of the generator model computes log-likelihood over the observed examples. The gradient is computed as

$$\frac{\partial}{\partial \theta} L_p(\theta) = \mathbb{E}_{p_{\text{data}}(\mathbf{x})p_\theta(\mathbf{z}|\mathbf{x})}\left[\frac{\partial}{\partial \theta} \log p_\theta(\mathbf{x}, \mathbf{z})\right]$$

**Limitation.** The MLE learning requires sampling from the generator posterior, which can be done by MCMC sampling as,

$$\mathbf{z}_{\tau+1} = \mathbf{z}_\tau + s\frac{\partial}{\partial \mathbf{z}_\tau} \log p_\theta(\mathbf{z}_\tau|\mathbf{x}) + \sqrt{2s}U_\tau$$

**However**, noise-initialized Langevin dynamics can be ineffective in traversing the latent space and hard to mix.

## Proposed Method

**Inference Model.** The generator model can be utilized as the **initializer** model for the MCMC sampling of EBM, while for the MCMC sampling of the generator posterior, an inference model is thus introduced,

$$q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mu_\phi(\mathbf{x}), V_\phi(\mathbf{x}))$$

which aims to initialize the MCMC generator posterior sampling.

**Joint Density.** With the EBM, generator and inference model, joint densities can be formulated as

$$P_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})$$
$$\Pi_{\alpha,\phi}(\mathbf{x}, \mathbf{z}) = \pi_\alpha(\mathbf{x})q_\phi(\mathbf{z}|\mathbf{x})$$
$$Q_\phi(\mathbf{x}, \mathbf{z}) = p_{\text{data}}(\mathbf{x})q_\phi(\mathbf{z}|\mathbf{x})$$

**Dual-MCMC Teaching.** In addition, we introduce two joint densities that incorporate the MCMC sampling as revision processes,

$$\tilde{P}_{\theta,\alpha}(\mathbf{x}, \mathbf{z}) = \mathcal{T}_\alpha^\mathbf{x} p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z}) \quad \tilde{Q}_{\phi,\theta}(\mathbf{x}, \mathbf{z}) = p_{\text{data}}(\mathbf{x})\mathcal{T}_\theta^\mathbf{z} q_\phi(\mathbf{z}|\mathbf{x})$$

**Learning energy-based model.** Therefore, learning the EBM is based on the minimization of KL divergences as

$$\text{KL}(\tilde{Q}_{\phi_t,\theta_t}(\mathbf{x}, \mathbf{z})\|\Pi_{\alpha,\phi}(\mathbf{x}, \mathbf{z})) - \text{KL}(\tilde{P}_{\theta_t,\alpha_t}(\mathbf{x}, \mathbf{z})\|\Pi_{\alpha,\phi}(\mathbf{x}, \mathbf{z}))$$
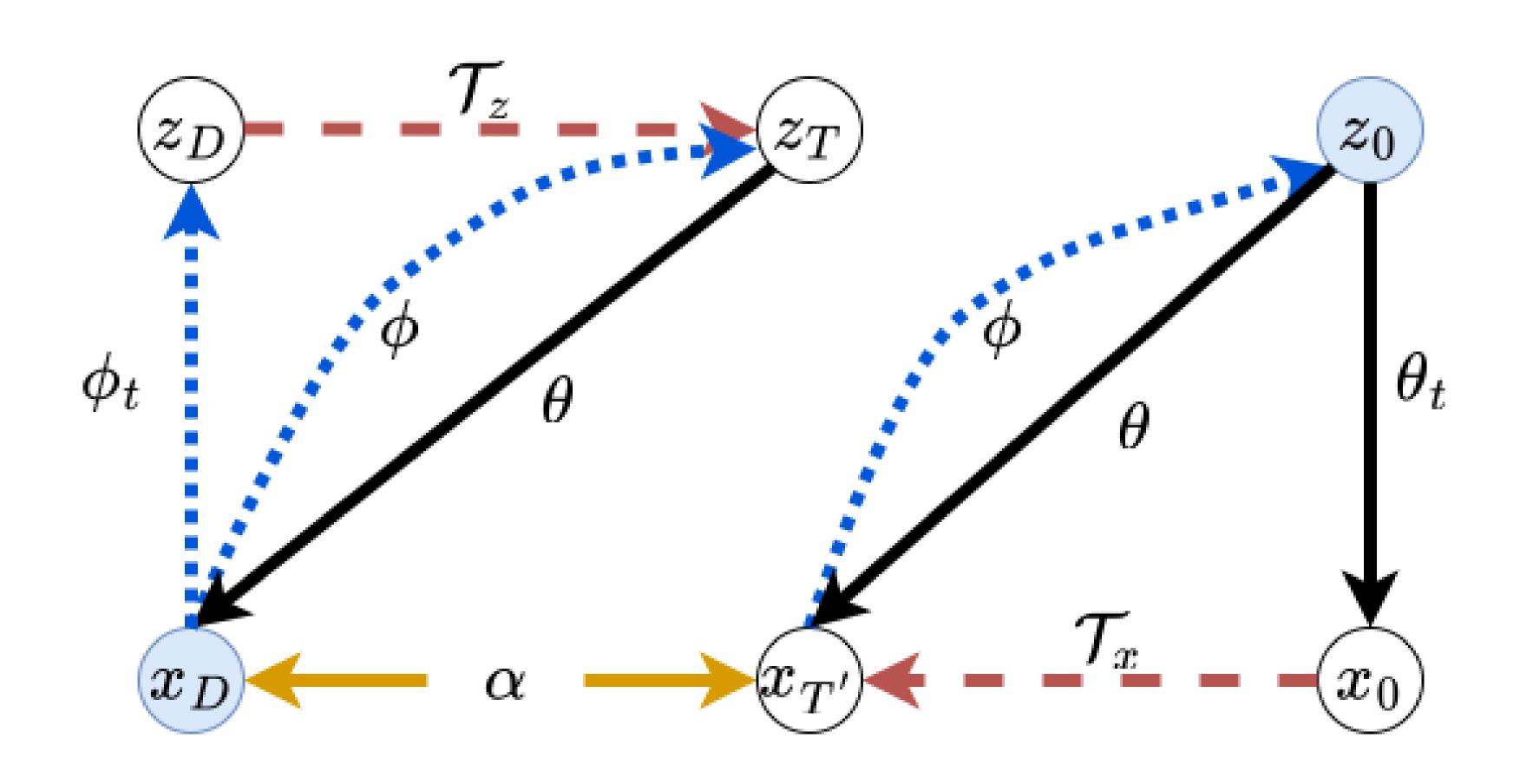
**Learning generator model.** The generator model is learned through the minimization of KL divergences as

$$\text{KL}(\tilde{Q}_{\phi_t,\theta_t}(\mathbf{x}, \mathbf{z})\|P_\theta(\mathbf{x}, \mathbf{z})) + \text{KL}(\tilde{P}_{\theta_t,\alpha_t}(\mathbf{x}, \mathbf{z})\|P_\theta(\mathbf{x}, \mathbf{z}))$$

**Learning inference model.** The generator model is learned through the minimization of KL divergences as

$$\text{KL}(\tilde{Q}_{\phi_t,\theta_t}(\mathbf{x}, \mathbf{z})\|Q_\phi(\mathbf{x}, \mathbf{z})) + \text{KL}(\tilde{P}_{\theta_t,\alpha_t}(\mathbf{x}, \mathbf{z})\|\Pi_{\alpha,\phi}(\mathbf{x}, \mathbf{z}))$$
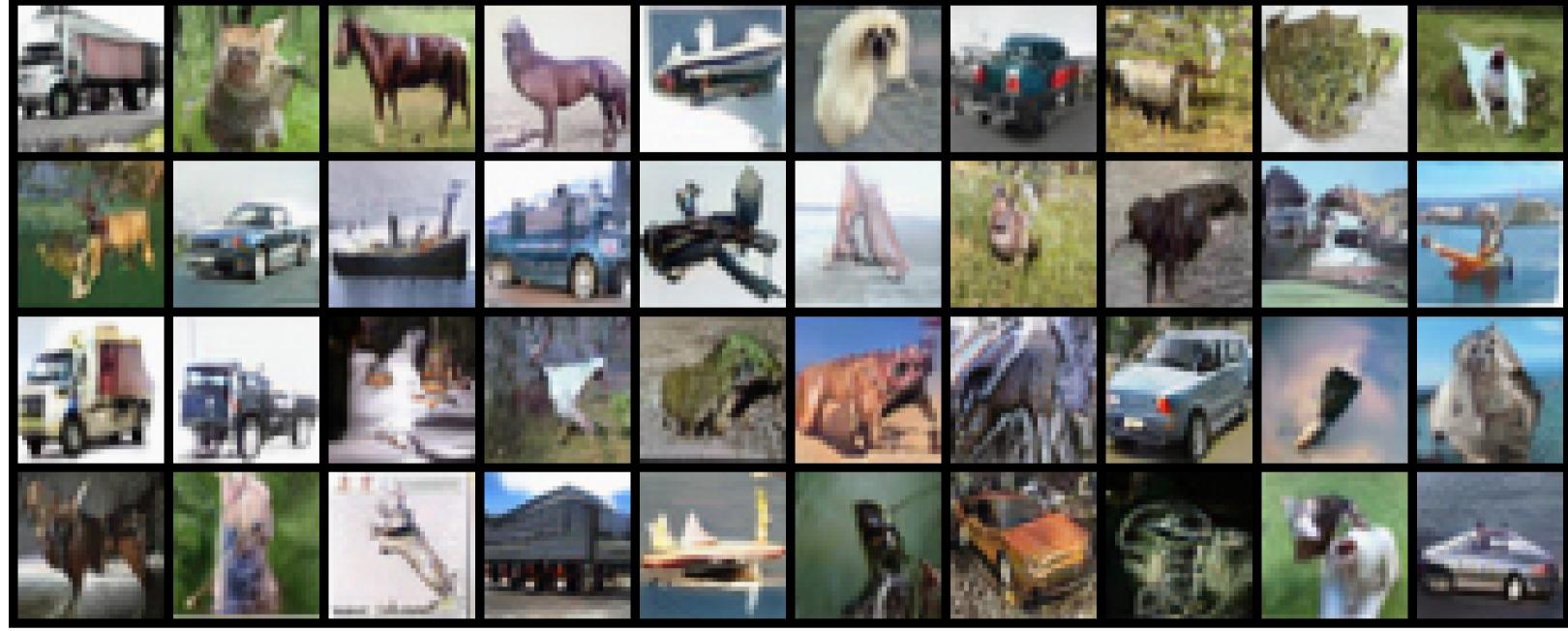
**Illustration for Dual-MCMC Teaching.**



## Experiment: Image Modelling

| Methods | CIFAR-10 | | CelebA-64 |
|---|---|---|---|
| | IS ($\uparrow$) | FID ($\downarrow$) | FID ($\downarrow$) |
| **Ours** | **8.55** | **9.26** | **5.15** |
| Cooperative EBM | 6.55 | 33.61 | 16.65 |
| Amortized EBM | 6.65 | - | - |
| Divergence Triangle | 7.23 | 30.10 | 18.21 |
| No MCMC EBM | - | 27.5 | - |
| Short-run EBM | 6.21 | - | 23.02 |
| IGEBM | 6.78 | 38.2 | - |
| ImprovedCD EBM | 7.85 | 25.1 | - |
| Diffusion EBM | 8.30 | 9.58 | 5.98 |
| VAEBM | 8.43 | 12.19 | 5.31 |

| Methods | CelebA-HQ-256 | LSUN-Church-64 |
|---|---|---|
| **Ours** | **15.89** | **4.56** |
| Diffusion EBM | - | 7.02 |
| VAEBM | 20.38 | 13.51 |
| NCP-VAE | 27.79 | - |
| GLOW | 68.93 | 59.35 |
| PGGAN | 21.7 | 6.1 |
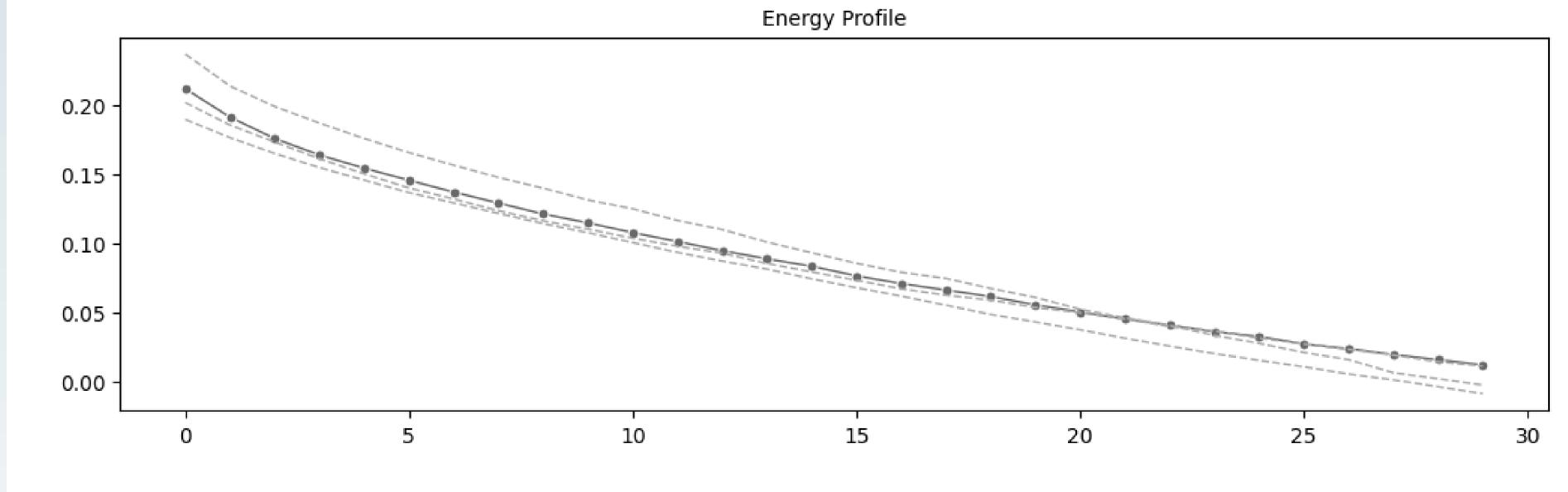


## Experiment: MCMC Revision

**Visualize of Langevin Transition on x.**
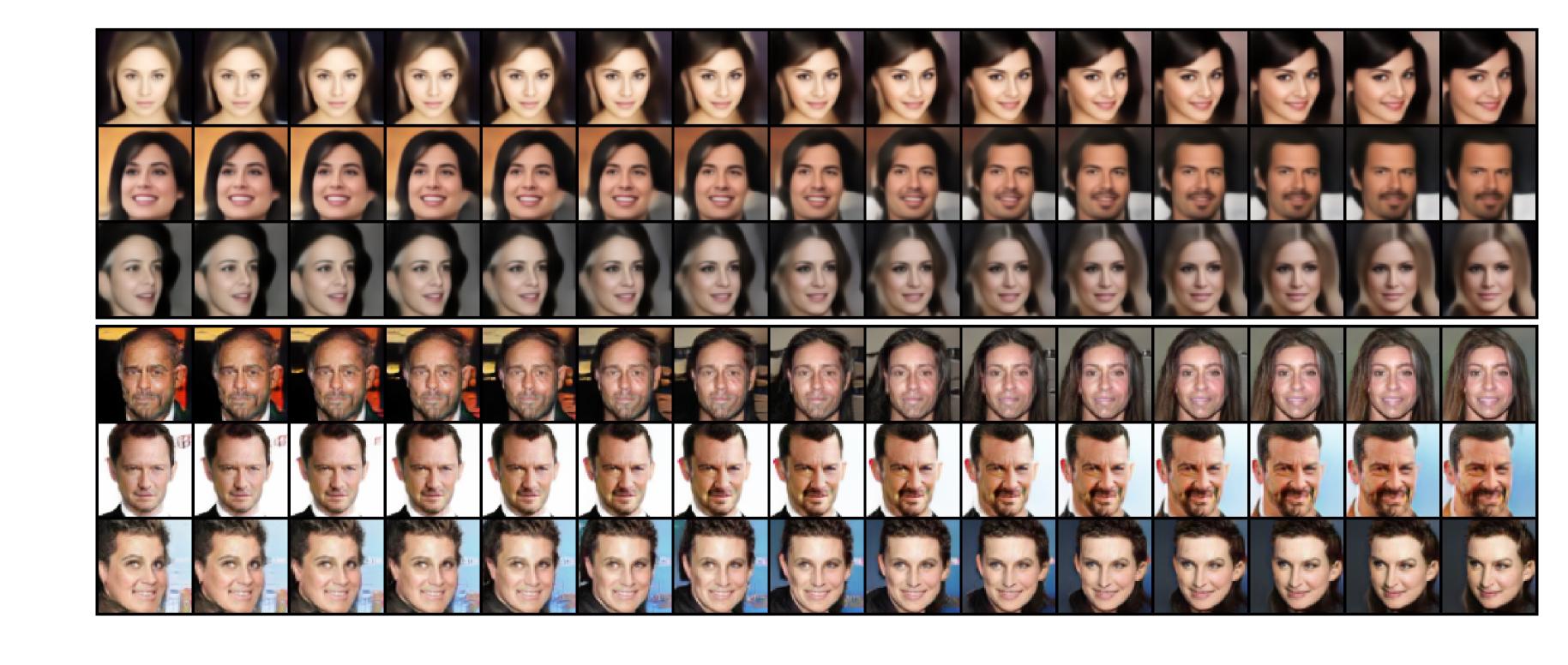


Energy Profile

**MCMC revision on x.** The leftmost images are sampled from the generator model, and the rightmost images are at the final step of the EBM-guided MCMC sampling. **Bottom:** Energy profile over steps. Only minor changes can be observed during the transition, suggesting that the generator has matched the EBM-guided MCMC revision.

## Experiment: Inference and Generator Model

**Image Reconstruction.**

| Methods | CIFAR-10 | CelebA-64 |
|---|---|---|
| VAE | 0.0341 | 0.0438 |
| ABP | 0.0183 | 0.0277 |
| Cooperative EBM | 0.0271 | 0.0387 |
| Divergence Triangle | 0.0237 | 0.0281 |
| Ours (Inf) | 0.0214 | 0.0227 |
| **Ours (Inf+L=10)** | **0.0072** | **0.0164** |

**Interpolation on Latent Space.**



The *top* and *bottom* three rows indicate image generation and reconstruction, respectively.