Convex Potential Mirror Langevin Algorithm for Efficient Sampling of Energy-Based Models

Zitao Yang[1], **Amin Ullah**[2], **Shuai Li**[3], **Li Fuxin**[4], **Jun Li**[1]
[1]Fudan University, [2]Boeing Research and Technology
[3]University of Bonn, [4]Oregon State University
yangzt21@m.fudan.edu.cn, amin.ullah@boeing.com, lishuai@iai.uni-bonn.de
lif@oregonstate.edu, jun_li@fudan.edu.cn

Abstract

This paper introduces the Convex Potential Mirror Langevin Algorithm (CPMLA), a novel method to improve sampling efficiency for Energy-Based Models (EBMs). CPMLA uses mirror Langevin dynamics with a convex potential flow as a dynamic mirror map for EBM sampling. This dynamic mirror map enables targeted geometric exploration on the data manifold, accelerating convergence to the target distribution. Theoretical analysis proves that CPMLA achieves exponential convergence with vanishing bias under relaxed log-concave conditions, supporting its efficiency in adapting to complex data distributions. Experiments on benchmarks like CIFAR-10, SVHN, and CelebA demonstrate CPMLA's improved sampling quality and inference efficiency over existing techniques.

1 Introduction

Energy-based models (EBMs) represent a class of generative machine learning models designed to capture and synthesize complex data distributions. EBMs define an unnormalized probability distribution via an energy function, assigning low energy values to likely data samples (corresponding to the target distribution) and high energy values to unlikely ones [55, 39, 10]. Known for their conceptual simplicity and training stability, EBMs have found diverse applications ranging from 3D object recognition [15] and analysis [56] to image segmentation [24], super-resolution restoration [60], machine translation [47], and protein folding [48, 52].

A critical limitation of energy-based models (EBMs) lies in their reliance on Markov Chain Monte Carlo (MCMC) sampling methods, particularly when operating in high-dimensional data spaces [4, 10, 29]. MCMC algorithms like Langevin dynamics often get trapped in local energy minima when the underlying data manifold is characterized by multi-modal energy landscapes [21] or exhibits non-Euclidean geometry [59, 21]. When sampling from the complex, highly multi-modal energy landscapes characteristic of deep EBMs, these MCMC methods can become computationally intensive and yield biased sampling [58]. These factors hinder the efficient approximation of complex distributions and can lead to slow convergence towards the target distribution.

Recent methods address sampling inefficiencies within EBMs. Some strategies refine MCMC initialization [18, 10], while others explore gradient approximation techniques [25, 29]. However, persistent challenges such as non-mixing issues remain unresolved [58]. Mirror Langevin algorithms have recently emerged as an alternative approach to alter sampling geometry via a fixed mirror map, i.e., a predefined function. Prior work [1, 30] demonstrates that mirror Langevin algorithms, under certain assumptions, exhibit vanishing bias (bias $\to 0$ as the step size $h \to 0$). This property ensures reliable convergence to the target distribution and improves sampling accuracy. Moreover, mirror Langevin algorithms achieve mixing times independent of the domain's condition number, enabling fast convergence [19, 59]. However, fixed mirror maps in conventional mirror Langevin algorithms

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

struggle to capture complex data manifolds efficiently, limiting their use for large-scale problems, especially those associated with deep neural networks.

This paper introduces Convex Potential Mirror Langevin Algorithm (CPMLA), a novel approach for sampling EBMs with enhanced efficiency. Unlike conventional mirror Langevin algorithms in Euclidean space, CPMLA employs a learnable, data-driven mirror map that actively infers the intrinsic manifold structure of the data. By parameterizing the mirror map as the gradient of a convex potential function (cf. Brenier's theorem [44]), CPMLA dynamically reorients sampling trajectories to align with the non-Euclidean geometry of the target distribution, enabling adaptive exploration of high-density regions while avoiding metastable states.

We employ a cooperative learning strategy that jointly trains the dynamic mirror map and the EBM. First, the dynamic mirror map is learned by optimizing a convex potential flow (CP-Flow) [20]. Building on Brenier's theorem for optimal transport [44], this formulation guarantees that CP-Flow – defined as the gradient of a convex potential function [3] – inherently captures the geometric structure of the data distribution. Then, the EBM is trained by contrasting the energy of real samples with that of those synthesized via CPMLA. Concurrently, synthesized samples are fed back into the CP-Flow training phase. This alternating process aligns the CP-Flow's transport dynamics with the EBM's energy-based density estimation, mitigating sampling bias and accelerating sampling convergence.

We theoretically analyze the convergence of CPMLA. Based on the recent study [21], we prove exponential convergence under relaxed log-concavity assumptions with two improvements. First, we specialize our proof for the dynamic mirror map modeled with deep neural networks. Second, beyond the sampling algorithm's error, our analysis also incorporates the approximation errors from modeling both the CP-Flow and the EBM with deep neural networks. These improvements broaden the applicability to a wider range of target distributions in various machine learning tasks. To the best of our knowledge, this is the first analysis of mirror Langevin algorithms within the framework of deep neural networks, resulting in exponential convergence with vanishing bias (Theorem 4.5).

We evaluate CPMLA across several benchmark datasets, including CIFAR-10, SVHN, and CelebA. The results demonstrate that CPMLA not only achieves superior sampling quality but also exhibits enhanced inference efficiency compared to existing cooperative algorithms. Specifically, CPMLA achieves an FID score 73% lower than Flow+EBM [13, 38], indicating a substantial improvement in visual quality. Additionally, CPMLA not only achieves a lower FID score (20.85 vs. 21.16) than CoopFlow [58] with fewer inference iterations (20 vs. 30) and less time (15.92s vs. 16.84s) as shown in Table 2, but also operates with only 0.9% of the parameter count w.r.t. the flow part as shown in Table 3, underscoring its efficiency in both sampling and inference. CPMLA also excels in specialized tasks like image reconstruction and inpainting, further emphasizing its effectiveness in tackling complex image processing challenges.

Our main contributions are summarized as follows:

- We propose a novel Convex Potential Mirror Langevin Algorithm (CPMLA) for efficient sampling of EBMs. The efficiency comes from the modification of the sampling geometry through a dynamic mirror map modeled with a deep neural network.
- We provide a theoretical convergence analysis of the proposed CPMLA, specifically focusing on deep neural networks under relaxed assumptions.
- We evaluate the efficacy of our proposed algorithm through comprehensive experimental analyses on various benchmark datasets, including CIFAR-10, SVHN, and CelebA. Our experiments demonstrate that our CPMLA achieves superior sampling efficiency compared to existing methods. Furthermore, it surpasses alternative approaches in terms of sample quality and the fidelity of image reconstruction and inpainting.

2 Background

2.1 Energy-Based Models

Energy-Based Models (EBMs) characterize a probability density over data $x \in \mathbb{R}^d$ as follows:

$$p_\theta(x) = \frac{1}{Z(\theta)} \exp[f_\theta(x)] \tag{1}$$

Here, $f_\theta : \mathbb{R}^d \to \mathbb{R}$ represents the negative energy function, parameterized by a neural network with parameters θ. The term $Z(\theta) = \int \exp[f_\theta(x)]dx$ is the normalizing constant, which is generally intractable to compute.

Generating samples from $p_\theta(x)$ involves Markov Chain Monte Carlo (MCMC) methods, with Langevin Monte Carlo (LMC) [50] being a prevalent choice. The LMC update rule is given by:

$$\hat{x}^{t+1} = \hat{x}^t + \frac{\delta^2}{2}\nabla_x f_\theta(\hat{x}^t) + \delta\varepsilon^t \tag{2}$$

where $\hat{x}^t$ is the sample at step t, δ is the step size, $\varepsilon^t \sim \mathcal{N}(0, I)$ is Gaussian noise, and the process is often initialized with $\hat{x}^0$ drawn from a simple distribution like uniform $p_0(x)$.

The parameters θ of the energy function are learned by maximizing the log-likelihood of observed data samples $x_i, i = 1, \ldots, n$ drawn from the true data distribution $p_{\text{data}}(x)$. The gradient of the log-likelihood objective is:

$$\nabla_\theta \log p_\theta(x) = \mathbb{E}_{p_{\text{data}}}\left[\nabla_\theta f_\theta(x)\right] - \mathbb{E}_{p_\theta}\left[\nabla_\theta f_\theta(x)\right] \approx \frac{1}{n}\sum_{i=1}^{n}\nabla_\theta f_\theta\left(x_i\right) - \frac{1}{n}\sum_{i=1}^{n}\nabla_\theta f_\theta\left(\hat{x}_i\right) \tag{3}$$

In this expression, $\hat{x}_i$ represents samples drawn from the current model distribution $p_\theta(x)$, usually obtained via LMC as described above. The expectation under p_θ, which implicitly depends on the intractable $Z(\theta)$, is estimated using these generated samples $\hat{x}_i$. Consequently, the learning updates θ by contrasting the average gradient of the energy function evaluated on real data with the average gradient evaluated on samples generated by the model.

2.2 Convex Potential Flow

A foundational requirement for CPMLA to satisfy the mirror Langevin algorithm is that the dynamic mirror map must be derived from a strongly convex potential function via its gradient. To this end, we choose Convex Potential Flow (CP-Flow) [20] for this role precisely because its architecture, based on Input-Convex Neural Networks (ICNNs), guarantees this convexity property. A standard normalizing flow, in contrast, does not generally have a convex potential, making it unsuitable for a mirror Langevin framework. As shown below, CP-Flow learns a tractable probability density by approximating the optimal transport map between a noise distribution and the target data distribution, specifically minimizing the quadratic cost (Monge) problem.

Optimal Transport The Monge problem [49] seeks an optimal transport map g minimizing the expected cost as follows:

$$J_c(p_X, p_Y) = \inf_{g:g(x)\sim p_Y} \mathbb{E}_{X\sim p_X}[c(x, g(x))] \tag{4}$$

where $c(x, y)$ is the given cost function.

Theorem 2.1. ***(Brenier's Theorem** [44]) Suppose μ and ν are probability measures with finite second moments, and assume that μ has a Lebesgue density p_X. In this case, there exists a convex potential G such that the gradient map $g = \nabla G$ (uniquely defined except for a null set) provides the solution to the Monge problem in Equation 4 with square cost function $c(x, y) = ||x - y||^2$.*

To approximate the optimal solution for the Monge problem, the convex potential is modeled with several layers of an input-convex neural network (ICNN) G_ϑ [3], which is convex w.r.t the input:

$$\begin{aligned} G_\vartheta(x) &= L^+_{K+1}\left(s\left(z_K\right)\right) + L_{K+1}(x) \\ z_k &:= L^+_k\left(s\left(z_{k-1}\right)\right) + L_k(x), \quad z_1 := L_1(x) \end{aligned} \tag{5}$$

where ϑ denotes parameters of the neural network, $L(x)$ denotes a linear layer, $L^+(x)$ denotes a linear layer with positive weights, and s is a non-decreasing convex activation function.

To ensure G_ϑ is strongly convex, which is required for ∇G_ϑ to be an invertible mirror map, a quadratic term is added: $G_\alpha(x) = G_\vartheta(x) + (\alpha/2)\|x\|_2^2$. For a small positive scalar α, this guarantees that the Hessian $\nabla^2 G_\alpha \succeq \alpha I \succ 0$. This modification ensures that the gradient ∇G_α is bijective and its inverse can be computed efficiently. For brevity, we omit the subscript α and use G_ϑ to denote the strongly convex potential hereafter.

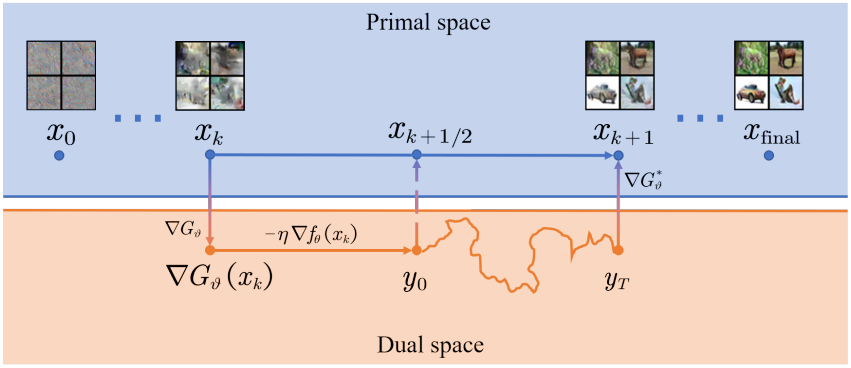


Figure 1: Overview of CPMLA sampling. Starting from a noisy sample x_0, CPMLA iteratively refines it by alternating between the primal space (interpretable images) and the dual space (geometry encoded by ∇G_ϑ). At each step, x_k is mapped to the dual space, then updated via an EBM energy gradient step and perturbed with geometry-aware noise scaled by $\nabla^2 G_\vartheta(x_k)$. Finally, the result is mapped back to the primal space as x_{k+1}, yielding progressively sharper samples that efficiently explore the data manifold.

Like standard flow-based models, CP-Flow is trained by maximizing the log-likelihood of the model density. This requires computing the log-determinant of the Hessian matrix of the convex potential, $\log\det H$, where $H = \nabla^2 G_\vartheta(x)$. For high-dimensional data, forming and storing the full Hessian is computationally infeasible. To overcome this, we use a matrix-free approach based on Hutchinson's trace estimator, which relies on efficient Hessian-vector products (HVPs). The gradient of the log-determinant can be estimated as:

$$\frac{\partial}{\partial \vartheta} \log\det H = \mathbb{E}_v \left[v^\top H^{-1} \frac{\partial H}{\partial \vartheta} v \right] \tag{6}$$

Algorithm 1 CP-Flow training objective

Procedure: Obj(G_ϑ, x, CG)
Sample Rademacher r
def hvp(v):
  return $v^\top \frac{\partial}{\partial x} \nabla G_\vartheta(x)$
$z \leftarrow$ stop_grad(CG(hvp, r))
return hvp$(z)^\top r$

where v is a random vector with zero mean and unit covariance (e.g., Rademacher). The term $H^{-1}v$ is expensive to compute directly. Instead, we reframe its calculation as a quadratic optimization problem, $z^* = \arg\min_z \left\{ \frac{1}{2} z^\top H z - v^\top z \right\}$, which can be solved efficiently using the conjugate gradient (CG) algorithm without ever instantiating H. This procedure is summarized in Algorithm 1.

3 Algorithms

3.1 Mirror Langevin Algorithm

To generate synthesized examples from a target distribution $p(x)$ with mirror Langevin dynamics [19], we solve the stochastic differential equation:

$$\begin{aligned} dy_t &= \nabla \log p(x_t) dt + \sqrt{2\nabla^2 G\left(x_t\right)} dW_t, \\ x_t &= \nabla G^*\left(y_t\right) \end{aligned} \tag{7}$$

where x_t and y_t are stochastic processes in the primal and dual spaces, respectively, W_t is the standard Brownian motion in $\mathbb{R}^d$, and ∇G is the mirror map. The term ∇G^* is the gradient of the convex conjugate G^*, which serves as the inverse of the mirror map, i.e., $(\nabla G)^{-1} = \nabla G^*$ (Appendix E).

We use the *Alternative Forward Discretization Scheme* (MLA_{AFD}) which has exponential convergence and vanishing bias [21, 1]. We use our CP-Flow ∇G_ϑ (Section 2.2) as the dynamic mirror map. For

an iteration with step size η, the update is:

$$\begin{aligned} x_{k+1/2} &\overset{1}{=} \nabla G^*_\vartheta\left(\nabla G_\vartheta\left(x_k\right) - \eta \nabla f_\theta\left(x_k\right)\right) \\ \text{solve } dy_t &= \sqrt{2\left[\nabla^2 G^*_\vartheta\left(y_t\right)\right]^{-1}} dW_t \\ &\overset{*}{=} \sqrt{2\nabla^2 G_\vartheta\left(\nabla G^*_\vartheta y_t\right)} dW_t \text{ for } y_0 \overset{2}{=} \nabla G_\vartheta\left(x_{k+1/2}\right) \\ x_{k+1} &= \nabla G^*_\vartheta\left(y_T\right) \end{aligned} \tag{8}$$

The $*$ step is derived from the property of convex conjugate [2] (see Appendix F). The computation of ∇G^*_ϑ in step 1 and ∇G_ϑ in step 2 can be simplified by noting that they are inverses and cancel each other out in successive iterations.

3.2 CPMLA

Our CPMLA facilitates exploration of the underlying data manifold. It achieves this by using MLA_{AFD} in Equation (8) with a CP-Flow dynamic mirror map, which dynamically transforms the sampling geometry based on the metric induced by the $\nabla^2 G_\vartheta$.

Like standard mirror Langevin methods, CPMLA alternates between updates in primal and dual spaces. The alternating sampling process entails transitioning between updating samples in the dual space using a dynamic mirror map for LMC exploration, followed by mapping the sample back to the primal space utilizing the inverse of the mirror map.

Figure 1 illustrates the sampling process of the proposed CPMLA. Specifically, each CPMLA iteration involves three steps: First, noise examples $\{y_0\}$ are generated from a standard Gaussian distribution $\mathcal{N}(0, I)$ in the dual space. And for each sampling step k, a noise vector ξ_k is generated from a Gaussian distribution $\mathcal{N}(0, \nabla^2 G_\vartheta(x_k))$. Second, starting from $\{y_0\}$, T steps of EBM sampling (gradient and SDE steps) are performed in the dual space, yielding $\{y_T\}$. Third, the inverse map transforms $\{y_T\}$ back to the primal space, yielding $\{\hat{x}\}$. The synthesized examples $\{\hat{x}\}$ are considered as outputs sampled by CPMLA.

Algorithm 2 shows the *cooperative learning* of EBM and CP-Flow. At each update, we re-initialize the MCMC chain. This is a standard practice in methods like Persistent Contrastive Divergence (PCD) to prevent chain collapse and ensure that model gradients are estimated from samples of the current model distribution, avoiding feedback from stale samples. First, we update the parameters ϑ of the CP-Flow using both original examples $\{x\}$ and synthesized examples $\{\hat{x}\}$. Then, we update the parameters θ of the EBM based on the contrast between $\{x\}$ and $\{\hat{x}\}$, as in Equation 3. The updates for both θ and ϑ are performed using the Adam optimizer, with learning rates and other hyperparameters specified in Appendix I. This cooperative mechanism simultaneously improves sampling efficiency and model expressiveness, creating a virtuous cycle of mutual enhancement.

In Algorithm 2, initial samples $\{y_0\} \sim \mathcal{N}(0, I)$ are drawn in the dual space, so no initial mapping with ∇G_ϑ is needed. We also use a computational trick to avoid the expensive matrix square root of the Hessian: the term $\sqrt{2\eta \nabla^2 G_\vartheta(x)} \cdot \tilde{\xi}_k$ (where $\tilde{\xi}_k \sim \mathcal{N}(0, I)$) is statistically equivalent to $\sqrt{2\eta} \cdot \xi_k$ (where $\xi_k \sim \mathcal{N}(0, \nabla^2 G_\vartheta(x))$). In practice, we approximate $\nabla^2 G_\vartheta$ with its diagonal to reduce computational complexity from $O(d^3)$ to $O(d)$, enabling efficient high-dimensional sampling.

4 Theoretical Analysis

This section presents the convergence analysis of CPMLA, the first for mirror Langevin algorithms with deep neural network mirror maps. Our analysis relies on standard properties of neural networks (e.g., bounded gradients via clipping) and a set of theoretical assumptions, which are standard in the analysis of Langevin-type algorithms [21, 1]. We provide detailed justifications for their validity in our framework below.

Assumption 4.1. (β-Mirror Log-Sobolev Inequality, β-Mirror LSI) The target distribution π satisfies β-Mirror LSI with constant w.r.t a given mirror map ∇G, i.e., for every locally lipschitz function h, it holds that π satisfies

$$\frac{2}{\beta} \int \|\nabla h\|^2_{[\nabla^2 G]^{-1}} d\pi \geq \int h^2 \log h^2 d\pi - \left(\int h^2 d\pi\right) \log \left(\int h^2 d\pi\right) \tag{9}$$

Algorithm 2 Convex Potential Mirror Langevin Algorithm (CPMLA)

Input: (1) Observed images $\{x\} \sim p_{\text{data}}(x)$; (2) Number of Mirror Langevin steps T; (3) Step size in dual space η.
Output: Parameters of EBM and CP-Flow $\{\theta, \vartheta\}$
Randomly initialize θ and ϑ.
repeat
  Sample noise examples $\{y_0\} \sim \mathcal{N}(0, I)$ in dual space.
  for $k = 0$ to $T - 1$ **do**
    Let $x_k = \nabla G^*_\vartheta(y_k)$
    Sample noise $\xi_k \sim \mathcal{N}(0, \nabla^2 G_\vartheta(x_k))$
    $y_{k+1/2} = y_k - \eta \nabla f_\theta(x_k)$
    $y_{k+1} = y_{k+1/2} + \sqrt{2\eta} \cdot \xi_k$
  end for
  Map back to primal space $\hat{x} = \nabla G^*_\vartheta(y_T)$
  Starting from $\{\hat{x}\}$, update ϑ by Algorithm 1
  Given $\{x\}$ and $\{\hat{x}\}$, update θ with Equation 3
until converged

Justification: This is a foundational assumption about the properties of the target data distribution itself, relative to the geometry induced by the mirror map. While difficult to verify empirically for complex, high-dimensional data distributions, it is a standard and necessary assumption in the literature for proving the convergence of Langevin-type algorithms in non-Euclidean spaces [21, 1]. Our contribution focuses on the aspects of the algorithm we can control and verify.

Assumption 4.2. (ζ-Self-Concordance) There exists a constant $\zeta \geq 0$ such that the conjugate mirror map ∇G^* satisfies that $\forall y, u, s, v$,

$$\left|\nabla^3 G^*(y)[u, s, v]\right| \leq 2\zeta \cdot \left(u^\top \nabla^2 G^*(y) u\right)^{1/2} \cdot \left(s^\top \nabla^2 G^*(y) s\right)^{1/2} \cdot \left(v^\top \nabla^2 G^*(y) v\right)^{1/2} \tag{10}$$

Justification: This assumption bounds the third derivative of the potential function relative to its second derivative, ensuring the geometry does not change too abruptly. We empirically validate this assumption for our trained models on CIFAR-10. As direct computation of the third-order derivative tensor is infeasible, we employ a matrix-free validation approach. We estimate the Frobenius norms of the Hessian $\nabla^2 G_\vartheta(x)$ and five random directional third derivatives $\nabla^3 G_\vartheta(x)[\vec{v}]$ using Hutchinson's estimator, which relies on efficient Hessian-vector products. We then compute the proxy metric $\hat{\zeta}_{\text{proxy}} = \frac{\|\nabla^3 G_\vartheta(x)[\vec{v}]\|_F}{\|\nabla^2 G_\vartheta(x)\|_F^{1.5} + \epsilon}$. Across all training checkpoints, the value of $\hat{\zeta}_{\text{proxy}}$ consistently remains small and stable (in the range $[10^{-4}, 10^{-2}]$), providing strong empirical support that this assumption holds in practice.

Assumption 4.3. (L-Relative Lipschitz) For all x, it holds that $f : \mathbb{R}^d \mapsto \mathbb{R}$ is differentiable with

$$\|\nabla f(x)\|_{[\nabla^2 G(x)]^{-1}} \leq L \tag{11}$$

Justification: This assumption is satisfied in our framework due to standard deep learning practices. Our potential function G is designed to be strongly convex, meaning its Hessian $\nabla^2 G(x) \succeq \alpha I$ for some $\alpha > 0$. In practice, we use gradient clipping on the EBM, which ensures that $||\nabla f(x)||$ is bounded by a constant C. This directly leads to $||\nabla f(x)||_{[\nabla^2 G(x)]^{-1}} \leq (1/\sqrt{\alpha})||\nabla f(x)|| \leq C/\sqrt{\alpha}$. Thus, the assumption holds by setting $L = C/\sqrt{\alpha}$.

Assumption 4.4. (Weaker γ-Relative Smooth) For all $x, x' \in dom(G)$,

$$\|\nabla f(x) - \nabla f(x')\|_{[\nabla^2 G(x')]^{-1}} \leq \gamma \cdot \|\nabla G(x) - \nabla G(x')\|_{[\nabla^2 G(x')]^{-1}} \tag{12}$$

Justification: Similarly, the gradient of our EBM, ∇f, is Lipschitz with some constant L_f (determined by the network architecture and enforced by weight decay and gradient clipping). The potential G is also smooth. This allows us to bound the relative smoothness, and the assumption holds by setting $\gamma = L_f/\alpha$.

Theorem 4.5 (Convergence of CPMLA)**.** *Let d be the dimension of the data space. For any mirror map ∇G, define $M := \exp\left(2\zeta D/\sqrt{\alpha}\right)$, where $D := \max_{u,v} \|\nabla G(u) - \nabla G(v)\|_2$ is the diameter*

of the image of ∇G. Under Assumptions 4.1-4.4, for any $\delta > 0$, after $k \geq \tilde{\Omega}\left(M\gamma^2 d/\beta^2\delta\right)$ iterations with step size $h = O(\beta/\gamma^2 d)$, the total variation distance between the sampling distribution ρ_t and the data distribution p_{data} satisfies:

$$d_{TV}(\rho_t, p_{data}) < \delta$$

where $\tilde{\Omega}(\cdot)$ hides polylogarithmic factors, i.e. $f = \tilde{\Omega}(g) \iff \exists\, c > 0,\, n_0,\, p \in \mathbb{N}$ such that $f(n) \geq c \cdot \frac{g(n)}{(\log n)^p}, \forall n \geq n_0$. $\delta = \sqrt{\delta_1/2} + \delta_2 + \delta_3$, with δ_1, δ_2, δ_3 being small constants related to the convergence error of CPMLA, approximation errors from the EBM and CP-Flow respectively.

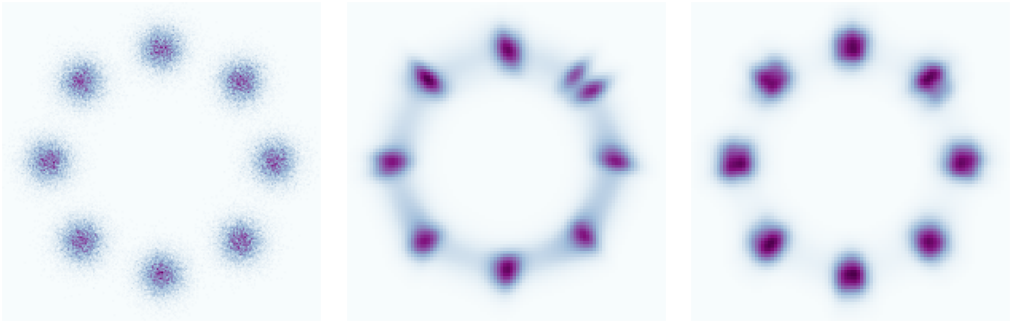
This theorem provides a non-asymptotic bound that characterizes the best achievable error of our framework. It states that if the EBM and the CP-Flow are trained to a certain approximation accuracy (represented by the epsilon terms), then the sampler is guaranteed to be within a certain Total Variation distance of the true data distribution. The number of iterations and step sizes are implicitly embedded in the conditions required to reach these error bounds.

Proof sketch: Lemma 1 from [21] provides the form of shifted drift and covariance of Equation 7 in primal space. Using this lemma, we express CPMLA in primal space as a weighted Langevin dynamics in differential form with a shifted drift term $\hat{\mu}$. Analyzing the Fokker-Planck equation for the conditional density $\rho_{t|0}(x_t \mid x_0)$, we bound the KL-divergence between ρ_t and the target π using integration by parts, the Cauchy-Schwarz inequality, and the mirror log-Sobolev inequality (Assumption 4.1). This yields a differential inequality showing exponential decay of the KL-divergence, with convergence rate governed by the algorithm's parameters and target distribution properties. The total variation bound $d_{TV}(\rho_t, p_{\text{data}}) < \delta$ decomposes into three terms: δ_1 measures the distance between the distribution ρ_k generated after k outer iterations of CPMLA (Algorithm 2) and the stationary distribution p_{θ^*} associated with the learned energy function f_{θ^*}. δ_2 and δ_3 represent the fundamental limitation in the expressive power of the chosen model architectures, namely the EBM f_θ and the CP-flow G_ϑ. They reflect how well the model family can intrinsically capture the target data distribution, irrespective of sampling or optimization efficiency.

The theorem states that, incorporating slight assumptions, CPMLA not only achieves exponential convergence but also exhibits vanishing bias, making it more applicable to the practical training scenario where both the energy model and mirror map parameters are continuously updated. Details on this proof may be found in Appendix H.

5 Experiments

We evaluate the proposed CPMLA on diverse tasks. We start with a toy example in Section 5.1. Next, we present image generation results in Section 5.2. Finally, we demonstrate CPMLA for image reconstruction and inpainting in Section 5.3.



(a) Data (b) CP-Flow (c) CPMLA

Figure 2: Comparison between CPMLA and CP-Flow for Fitting Eight Gaussians. CPMLA reaches the same result in just 3 iterations that CP-Flow takes 10 iterations to achieve.

5.1 Toy Model Study

We first illustrate our approach on a toy example. Specifically, We apply CPMLA to model the eight Gaussians density from [42] and [5]. The results, presented in Figure 2, show that CPMLA efficiently fits these distributions. It demonstrates that on synthetic data, CPMLA provides a reliable approximation of the target distribution without introducing bias. Notably, CPMLA matches CP-Flow's 10-iteration result [20] in only 3 iterations, highlighting its superior convergence speed.

5.2 Image Generation

Model type	**Models**	**FID**$\downarrow$
VAE	VAE [26]	78.41
Autoregressive	PixelCNN [43]	65.93
GAN	WGAN-GP [14]	36.40
	StyleGAN2-ADA [23]	2.92
Score-Based	NCSN [45]	25.32
	NCSN++ [46]	2.20
Flow	Glow [27]	45.99
	Residual Flow [6]	46.37
EBMs	LP-EBM [41]	70.15
	EBM-SR [39]	44.50
	EBM-IG [10]	38.20
	CoopVAEBM [57]	36.20
	CoopNets [54]	33.61
Flow+EBM	NT-EBM [38]	78.12
	EBM-FCE [13]	37.30
	CoopFlow (T=20) [58]	30.74
	CoopFlow (T=30) [58]	21.16
CPMLA (Ours)	CPMLAprt (T=20)	20.85
	CPMLA (T=30)	21.09

Table 1: FID scores on the CIFAR-10. Our work focuses on improving the sampling efficiency and quality for the EBM family of models, making them more competitive. While other classes of generative models like score-based diffusion (e.g., NCSN++) or flow-matching models may achieve lower (better) FID scores on benchmark datasets, a direct comparison is not the primary goal. EBMs offer greater modeling flexibility, as they only require specifying an unnormalized energy function, unlike models requiring specific architectures or tractable noise processes. Our method helps make this flexibility more practical by closing the sample quality gap. The comparison to CoopFlow, a strong EBM baseline, demonstrates CPMLA's superior efficiency in this context.

We evaluate image synthesis performance on three datasets: CIFAR-10 [28], which consists of 50,000 training images and 10,000 test images across 10 categories; SVHN [37], a dataset with over 70,000 training images and more than 20,000 test images of house numbers; and CelebA [31], a large dataset of celebrity faces containing over 200,000 images. For fair comparison, all images are resized to 32×32 pixels. We present results under two settings. *CPMLA*: CP-Flow and EBM trained from scratch. *CPMLAprt*: CP-Flow is first pretrained on data, then used to initialize CPMLA training. Pretraining provides a better initialization, potentially leading to higher quality images.

We present qualitative results (Figure 3) and quantitative FID scores (Table 1). FID scores [17] are computed based on 50,000 samples. Our models outperform most baselines, achieving lower FID scores compared to standalone normalizing flows and previous EBM+flow methods [13, 38].

The results demonstrate that CPMLA is parameter-efficient and effective compared to other cooperative and flow-only approaches. In particular, compared to CoopFlow, CPMLA provides a distinct

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

Figure 3: Generated Samples (32 × 32 pixels) by CPMLA from CIFAR-10, SVHN, and CelebA datasets. These images are produced under the CPMLAprt training setting.

Models	**Time (s/1k images)**	**FID↓**
CoopFlow (T=30)	16.84	21.16
CPMLAprt (T=20)	15.92	20.85

Table 2: Wall-clock time (s/1k images) and FID comparison on CIFAR-10 (50k samples). CPMLA achieves a lower FID than CoopFlow with fewer LMC iterations and less computation time.

	EBM part	Flow part
CoopFlow	17.13M	28.78M
CPMLA	17.13M	0.26M

Table 3: Comparison of the parameter amount between CoopFlow and CPMLA. CPMLA achieves lower FID scores to CoopFlow with only 0.9% parameter count w.r.t the flow part.

advantage in terms of inference efficiency. **(i)** As shown in Table 2, **CPMLA achieves a lower FID than CoopFlow with fewer LMC iterations and less computation time**. Figure 4 further illustrates how CPMLA's FID improves faster than CoopFlow's across sampling steps ($T = 3$ to $T = 30$), highlighting its superior convergence speed. **(ii)** CPMLA's CP-Flow component uses significantly fewer parameters (0.27M) than CoopFlow's normalizing flow (28.78M, see Table 3). Remarkably, **CPMLA outperforms CoopFlow while using only 0.9% of its parameters**.

(iii) To further analyze computational cost, we compare the total training time of CPMLA and CoopFlow on CIFAR-10 under realistic hardware constraints. When maximizing batch size to fit within 24GB of VRAM, CoopFlow is estimated to require approximately **38 hours** for training, whereas CPMLA completes in only **10.5 hours**. While a direct per-iteration comparison for a fixed batch size shows that CPMLA is marginally slower (15.7 s/iter vs. 12.0 s/iter for CoopFlow) due to the more complex CP-Flow architecture, its memory efficiency allows for larger batches, leading to significantly better overall training throughput. This highlights CPMLA's superior training efficiency in practical, resource-constrained scenarios.

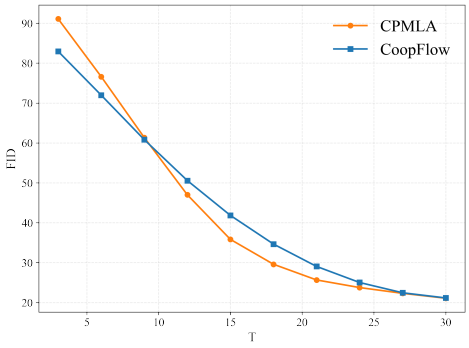


Figure 4: FID comparison from $T = 3$ to $T = 30$ between CPMLA and CoopFlow on CIFAR-10 dataset. From an inferior initialization, CPMLA demonstrates faster inference speeds than CoopFlow.

5.3 Image Reconstruction and Inpainting

We evaluate CPMLA for image reconstruction task, with a focus on the CIFAR-10 testing set as illustrated in Figure 6 (Appendix J). The high fidelity of reconstructions demonstrates the model's capability. This empirical evidence suggests the CPMLA framework can function effectively for reconstruction.

We further demonstrate CPMLA for image inpainting. Let's assume we have an image represented by a function $I : \Omega \subset \mathbb{R}^2 \to \mathbb{R}^3$, where Ω is the domain of the image, and $I(x, y)$ gives the color at coordinates (x, y). We optimize the objective energy function Equation 13 to measure the difference between the restored region and the original image. To ensure that the restoration process does not alter the undamaged parts of the original image, we introduce a constraint: $u(x, y) = I(x, y) \quad \text{if} \quad M(x, y) = 1$.

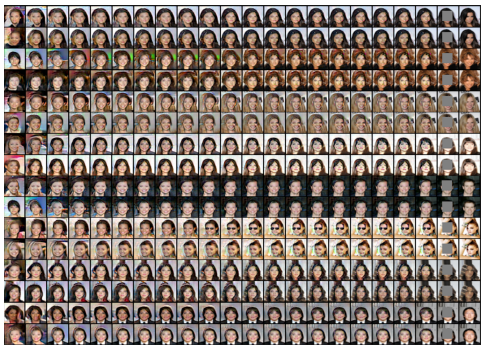


Figure 5: Image inpainting on the CelebA. The first 17 columns exhibit the inpainting results at various iterations, while the last two columns visually compare the masked images and the originals. CPMLA faithfully inpaints the masked images.

$$E(u) = \int_{\Omega} (I(x, y) - u(x, y))^2 \cdot M(x, y) dx dy \tag{13}$$

Experiments conducted on CelebA, are shown in Figure 5. The first 17 columns show inpainting results over optimization iterations, offering a dynamic view of the reconstruction process. The last two columns visually compare the masked images and the originals. Figure 5 shows CPMLA successfully inpaints masked images from diverse initializations.

6 Limitations and Future works

In our experiments, estimating the Hessian can introduce bias to the optimal point. However, compared to the exact evaluation of the inverse Hessian, this is a trade-off we must make. While our experimental results demonstrate effectiveness for diverse sampling tasks, the mirror LSI assumption (Assumption 4.1) is rather general, as we cannot ensure that the target distributions of different sampling tasks satisfy this assumption, particularly in EBMs where the target distribution is highly complex. We note that, like other generative models, improvements could potentially be misused (e.g., for deepfakes). For future work, we plan to explore the deeper connection between sampling and optimization. For instance, can optimization techniques (e.g., adaptive step sizes like Adam, trust regions) accelerate sampling or correct bias? Additionally, higher-order discretizations (e.g., Runge-Kutta) might improve convergence rates. We aim to investigate these questions and further advance the field of sampling and optimization.

7 Conclusions

This paper presented CPMLA, a sampling algorithm developed for Energy-Based Models (EBMs). The method utilizes Convex Potential Flow (CP-Flow) as a dynamic mirror map, allowing the sampling process to adapt to the underlying geometry of the data distribution. This adaptive mechanism facilitates sampling with vanishing bias and contributes to sampling efficiency. Theoretical analysis establishes the algorithm's convergence properties within the mirror Langevin dynamics framework. The algorithm demonstrated its applicability and effectiveness in image generation, reconstruction, and inpainting tasks. Experimental results indicated favorable performance concerning computational time and parameter count compared to related methods. In summary, CPMLA provides a principled approach to EBM sampling, integrating theoretical convergence properties with empirical performance. The method's capacity for adaptive sampling suggests its potential utility for enhancing the application of EBMs in various domains.

Acknowledgements

Zitao Yang and Jun Li are supported by the National Natural Science Foundation of China (No. 72342016). Amin Ullah and Fuxin Li are supported by ONR/NAVSEA contracts N0014-21-1-2052 and N00024-10-D-6318.

References

[1] Kwangjun Ahn and Sinho Chewi. Efficient constrained sampling via the mirror-langevin algorithm. *Advances in Neural Information Processing Systems*, 34:28405–28418, 2021.

[2] Shun-ichi Amari. *Information geometry and its applications*, volume 194. Springer, 2016.

[3] Brandon Amos, Lei Xu, and J. Zico Kolter. Input convex neural networks. *International Conference on Machine Learning*, 2017.

[4] Adrian Barbu, Song-Chun Zhu, et al. *Monte Carlo Methods*, volume 35. Springer, 2020.

[5] Jens Behrmann, Will Grathwohl, Ricky TQ Chen, David Duvenaud, and Jörn-Henrik Jacobsen. Invertible residual networks. In *International conference on machine learning*, pages 573–582. PMLR, 2019.

[6] Ricky TQ Chen, Jens Behrmann, David K Duvenaud, and Jörn-Henrik Jacobsen. Residual flows for invertible generative modeling. *Advances in Neural Information Processing Systems*, 32, 2019.

[7] Xiang Cheng, NiladriS. Chatterji, PeterL. Bartlett, and MichaelI. Jordan. Underdamped langevin mcmc: A non-asymptotic analysis. *Cornell University - arXiv,Cornell University - arXiv*, Jul 2017.

[8] Sinho Chewi, ThibautLe Gouic, Chen Lu, Tyler Maunu, Philippe Rigollet, and AustinJ. Stromme. Exponential ergodicity of mirror-langevin diffusions. *Neural Information Processing Systems,Neural Information Processing Systems*, Jan 2020.

[9] Arnak S. Dalalyan and Avetik Karagulyan. User-friendly guarantees for the langevin monte carlo with inaccurate gradient. *Stochastic Processes and their Applications*, 129(12):5278–5311, Dec 2019.

[10] Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. *Neural Information Processing Systems*, Jan 2019.

[11] Alain Durmus and Eric Moulines. High-dimensional bayesian inference via the unadjusted langevin algorithm. *Bernoulli,Bernoulli*, Dec 2016.

[12] Alain Durmus and Eric Moulines. Non-asymptotic convergence analysis for the unadjusted langevin algorithm. *Annals of Applied Probability,Annals of Applied Probability*, Dec 2016.

[13] Ruiqi Gao, Erik Nijkamp, DiederikP. Kingma, Zhen Xu, AndrewM. Dai, and Ying Wu. Flow contrastive estimation of energy-based models. *Cornell University - arXiv,Cornell University - arXiv*, Dec 2019.

[14] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

[15] Fredrik K Gustafsson, Martin Danelljan, and Thomas B Schon. Accurate 3d object detection using energy-based models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2855–2864, 2021.

[16] Tian Han, Yang Lu, Song-Chun Zhu, and Ying Nian Wu. Alternating back-propagation for generator network. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1), Jun 2022.

[17] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Neural Information Processing Systems,Neural Information Processing Systems*, Jan 2017.

[18] Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, page 1771–1800, Aug 2002.

[19] Ya-Ping Hsieh, Ali Kavis, Paul Rolland, and Volkan Cevher. Mirrored langevin dynamics. *neural information processing systems*, 2018.

[20] Chin-Wei Huang, Ricky T. Q. Chen, Christos Tsirigotis, and Aaron Courville. Convex potential flows: Universal probability distributions with optimal transport and convex optimization. *Learning*, 2020.

[21] Qijia Jiang. Mirror langevin monte carlo: the case under isoperimetry. *Neural Information Processing Systems*, 2021.

[22] Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the fokker-planck equation. *Siam Journal on Mathematical Analysis*, 1998.

[23] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in neural information processing systems*, 33:12104–12114, 2020.

[24] Ramgopal Kashyap and Pratima Gautam. Fast medical image segmentation using energy-based method. In *Pattern and data analysis in healthcare settings*, pages 35–60. IGI Global, 2017.

[25] Taesup Kim and Yoshua Bengio. Deep directed generative models with energy-based probability estimation. *Cornell University - arXiv*, Feb 2016.

[26] DiederikP. Kingma and Max Welling. Auto-encoding variational bayes. *arXiv: Machine Learning,arXiv: Machine Learning*, Dec 2013.

[27] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

[28] Alex Krizhevsky. Learning multiple layers of features from tiny images. Jan 2009.

[29] Rithesh Kumar, Anirudh Goyal, Aaron Courville, and Yoshua Bengio. Maximum entropy generators for energy-based models. *arXiv: Learning,arXiv: Learning*, Jan 2019.

[30] Ruilin Li, Molei Tao, Santosh S Vempala, and Andre Wibisono. The mirror langevin algorithm converges with vanishing bias. In *International Conference on Algorithmic Learning Theory*, pages 718–742. PMLR, 2022.

[31] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

[32] Yi-An Ma, Niladri S. Chatterji, Xiang Cheng, Nicolas Flammarion, Peter L. Bartlett, and Michael I. Jordan. Is there an analog of nesterov acceleration for gradient-based mcmc? *Bernoulli*, May 2021.

[33] Wenlong Mou, Nicolas Flammarion, Martin J. Wainwright, and Peter L. Bartlett. Improved bounds for discretization of langevin diffusions: Near-optimal rates without convexity. *Bernoulli*, 28(3), Aug 2022.

[34] Wenlong Mou, Yuanlin Ma, MartinJ. Wainwright, PeterL. Bartlett, and MichaelI. Jordan. High-order langevin diffusion yields an accelerated mcmc algorithm. *arXiv: Machine Learning,arXiv: Machine Learning*, Aug 2019.

[35] Yurii Nesterov et al. *Lectures on convex optimization*, volume 137. Springer, 2018.

[36] Yurii Nesterov and Arkadii Nemirovskii. Interior-point polynomial algorithms in convex programming, Jan 1994.

[37] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[38] Erik Nijkamp, Ruiqi Gao, Pavel Sountsov, V. Srinivas, Bo Pang, Song-Chun Zhu, and Ying Wu. Learning energy-based model with flow-based backbone by neural transport mcmc. *Cornell University - arXiv,Cornell University - arXiv*, Jun 2020.

[39] Erik Nijkamp, Mitch Hill, Song-Chun Zhu, and YingNian Wu. Learning non-convergent non-persistent short-run mcmc toward energy-based model. *Neural Information Processing Systems*, Jan 2019.

[40] Felix Otto and Cédric Villani. Generalization of an inequality by talagrand and links with the logarithmic sobolev inequality. *Journal of Functional Analysis*, 2000.

[41] Bo Pang, Tian Han, Erik Nijkamp, Song-Chun Zhu, and Ying Nian Wu. Learning latent space energy-based prior model. *Advances in Neural Information Processing Systems*, 33:21994–22008, 2020.

[42] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30, 2017.

[43] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017.

[44] Filippo Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55(58-63):94, 2015.

[45] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[46] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[47] Lifu Tu, Richard Yuanzhe Pang, Sam Wiseman, and Kevin Gimpel. Engine: Energy-based inference networks for non-autoregressive machine translation. *arXiv preprint arXiv:2005.00850*, 2020.

[48] Jérôme Tubiana, Simona Cocco, and Rémi Monasson. Learning protein constitutive motifs from sequence data. *arXiv: Quantitative Methods*, Mar 2018.

[49] Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.

[50] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688, 2011.

[51] Andre Wibisono. Proximal langevin algorithm: Rapid convergence under isoperimetry. *arXiv: Machine Learning,arXiv: Machine Learning*, Nov 2019.

[52] Jiaxiang Wu, Shitong Luo, Tao Shen, Haidong Lan, Sheng Wang, and Junzhou Huang. Ebm-fold: fully-differentiable protein folding powered by energy-based models. *arXiv preprint arXiv:2105.04771*, 2021.

[53] Jianwen Xie, Yang Lu, Ruiqi Gao, and Ying Nian Wu. Cooperative learning of energy-based model and latent variable model via mcmc teaching. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Jun 2022.

[54] Jianwen Xie, Yang Lu, Ruiqi Gao, Song-Chun Zhu, and Ying Nian Wu. Cooperative training of descriptor and generator networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 27–45, Jan 2020.

[55] Jianwen Xie, Yang Lu, Song-Chun Zhu, and Ying Nian Wu. A theory of generative convnet. *arXiv: Machine Learning*, 2016.

[56] Jianwen Xie, Zilong Zheng, Ruiqi Gao, Wenguan Wang, Song-Chun Zhu, and Ying Nian Wu. Learning descriptor networks for 3d shape synthesis and analysis. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8629–8638, 2018.

[57] Jianwen Xie, Zilong Zheng, and Ping Li. Learning energy-based model with variational auto-encoder as amortized sampler. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10441–10451, 2021.

[58] Jianwen Xie, Yaxuan Zhu, Jun Li, and Ping Li. A tale of two flows: Cooperative learning of langevin flow and normalizing flow toward energy-based model. *arXiv preprint arXiv:2205.06924*, 2022.

[59] Kelvin Shuangjian Zhang, Gabriel Peyré, Jalal M. Fadili, and Marcelo Pereyra. Wasserstein control of mirror langevin monte carlo. *Le Centre pour la Communication Scientifique Directe - HAL - Université de Nantes*, 2020.

[60] Zilong Zheng, Jianwen Xie, and Ping Li. Patchwise generative convnet: Training energy-based models from a single natural image for internal learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2961–2970, 2021.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims are made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of this work are discussed in Section 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The paper provides the full set of assumptions and a complete and correct proof.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper fully discloses all the information needed to reproduce the main experimental results of the paper (Appendix I).

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: This paper provides open access to the data (Section 5), but not to the code. We will provide an open resource of our code when this paper is accepted.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The paper specifies all the training and test details necessary to understand the results (Section I).

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The paper reports appropriate information about the statistical significance of the experiments (Section 5).

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provides sufficient information on the computer resources (Appendix I).

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper mentioned both potential positive societal impacts and negative societal impacts of the work performed (Section 6).

    Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [No]

    Justification: This is challenging.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The creators or original owners of assets used in the paper are properly credited. The license and terms of use explicitly mentioned and are properly respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: This paper uses LLM only for writing, editing and does not involve any important, original and non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Related Work

Langevin sampling Many discretizations of Langevin dynamics within Euclidean geometry have been studied in the literature, with non-asymptotic error bounds derived for various metrics like Kullback-Leibler divergence, Total Variation, and Wasserstein distance. The most extensively studied scenarios include cases where the target distribution is m-strongly log-concave [12, 11, 7, 9, 34] and those where it is relaxed log-concave [51, 32, 33].

Mirror Langevin Dynamics (MLD) extends standard Langevin dynamics by operating in a 'curved' Riemannian geometry defined by a convex potential, rather than the 'flat' Euclidean space. This allows the sampling process to adapt to the underlying geometry of the data distribution, which can lead to faster convergence. When the convex potential is quadratic, MLD reduces exactly to standard Langevin dynamics. MLD has recently gained attention in the field of non-Euclidean geometry sampling due to its superior convergence properties in constrained optimization problems. Introduced by [19] as a measure transformation of the classical Langevin dynamics, its convergence under relaxed log-concavity was investigated by [59], where the authors demonstrated convergence to a Wasserstein ball with non-vanishing bias. [1] showed vanishing bias under similar conditions as the step size decreases. [8] studied convergence using similar functional inequalities, but without exploring practical applications.

Cooperative learning The cooperative learning concept, first introduced in [53], involves the joint maximum likelihood training of a ConvNet-EBM [55] and a top-down generator [16]. Similarly, [57] replaced the generator in the original CoopNets with a variational autoencoder (VAE) [26] to improve inference efficiency. Our learning algorithm draws inspiration from the recent Coopflow approach [58], which collaboratively trains a Langevin flow and a normalizing flow to improve initial samples.

B Optimal Transport

In recent years, there has been increasing interest in applying optimal transport theory to generative modeling, which considers the training process as a task of minimizing the distance between two probability distributions. More specifically, the objective is to transform a random distribution into a target distribution that closely approximates the underlying data distribution, with the distance between these two distributions often quantified using the Wasserstein distance in the context of optimal transport. The Wasserstein p-distance between two probability measures μ and ν on a metric space M with finite p-moments is

$$W_p(\mu,\nu) = \left(\inf_{\gamma \in \Gamma(\mu,\nu)} \mathbf{E}_{(x,y)\sim\gamma} d(x,y)^p \right)^{1/p} \tag{14}$$

where $\Gamma(\mu,\nu)$ is the set of all couplings of μ and ν. [22] shows that the Langevin dynamics manifests as the gradient flow of the Kullback-Leibler divergence within the probability measure space, characterized by the Wasserstein metric, as elucidated through the Fokker-Planck equation. This observation establishes a more substantial linkage between the realms of sampling and optimization; see also by the paper [40].

C Mirror Langevin Algorithm for Constrained Sampling

The mirror Langevin algorithm is a powerful technique for sampling from complex distributions, particularly those with constraints or intricate geometries. It leverages the concept of mirror maps, which can adapt to the geometric structure of the target distribution, enabling efficient and accurate sampling. One notable application of the mirror Langevin algorithm is constrained sampling, where the goal is to draw samples from a population while adhering to specific conditions or constraints.

Constrained sampling involves drawing a set of samples S from a population $U = u_1, u_2, ..., u_n$ while satisfying predefined constraint conditions expressed as inequalities. The general form of these constraints can be written as:

$$C(x) : g_i(x) \leq 0, \quad i = 1, 2, ..., m \tag{15}$$

Here, $g_i(x)$ represents the constraint functions, and the goal is to ensure that $g_i(x) \leq 0$ for all i.

The mirror Langevin algorithm addresses constrained sampling by leveraging mirror maps that can adapt to the geometry of the constraints. Specifically, we employ CP-Flow as our dynamic mirror map, which utilizes Implicit Convex Neural Networks (ICNNs) to approximate arbitrary convex functions effectively. Given that the derivative of a convex function is monotonic, we can ensure that the convergence of potential functions implies the convergence of the associated gradient fields, as stated in the following theorem:

Theorem C.1 (Optimality (Theorem 4 in [20]))**.** *Let F be the Brenier potential of $X \sim \mu$ and $Y \sim \nu$, and let G_n be a convergent sequence of differentiable, convex potentials, such that $\nabla G_n \circ X \to Y$ in distribution. Then, ∇G_n converges almost surely to ∇F.*

In our context, where μ represents the unconstrained space and ν serves as the convex constraint, Theorem C.1 guarantees the existence of a convex potential whose derivative maps μ to ν. By leveraging the expressive nature of ICNNs through CP-Flow, we can adapt to the intrinsic geometry of the constraints, resulting in accelerated convergence during constrained sampling. The mirror Langevin algorithm's ability to handle complex constraints makes it a valuable tool for various applications beyond constrained sampling, such as sampling from EBMs, Bayesian inference, and more [1].

D Derivative of CP-Flow

[20] presents an alternative formulation of the gradient as the solution to a convex optimization problem, eliminating the need to differentiate through the log-determinant estimation process. By adapting the gradient formula from Appendix C in [6] to the context of convex potentials, and utilizing Jacobi's formula* alongside the adjugate representation of the matrix inverse †, we derive the following identity for any invertible matrix H parameterized by θ:

$$\frac{\partial}{\partial \theta} \log\det H = \frac{1}{\det H}\frac{\partial}{\partial \theta}\det H \overset{*}{=} \frac{1}{\det H}\operatorname{tr}\left(\operatorname{adj}(H)\frac{\partial H}{\partial \theta}\right) \overset{\dagger}{=} \operatorname{tr}\left(H^{-1}\frac{\partial H}{\partial \theta}\right) = \mathbb{E}_v\left[v^\top H^{-1}\frac{\partial H}{\partial \theta}v\right] \tag{16}$$

In the last equality, [20] apply the Hutchinson trace estimator using a Rademacher random vector v, which is an unbiased Monte Carlo gradient estimator.

E Property of Convex Conjugate

G^* is the convex conjugate of G. Then

$$\begin{aligned} \nabla G(x) = x^*(x) &:= \arg\sup_{x^*} \langle x, x^*\rangle - G^*\left(x^*\right) \\ \nabla G^*\left(x^*\right) = x\left(x^*\right) &:= \arg\sup_{x} \langle x, x^*\rangle - G(x) \end{aligned} \tag{17}$$

Hence

$$x = \nabla G^*(\nabla G(x)) \quad \text{and} \quad x^* = \nabla G\left(\nabla G^*\left(x^*\right)\right) \tag{18}$$

F Lemma of Convex Conjugate

Lemma F.1. *Suppose we have a dualistic structure*

$$\boldsymbol{\xi}^* = \nabla G(\boldsymbol{\xi}), \quad \boldsymbol{\xi} = \nabla G^*\left(\boldsymbol{\xi}^*\right) \tag{19}$$

G^* is the Legendre dual of G, which is defined as

$$G^*\left(\boldsymbol{\xi}^*\right) = \max_{\boldsymbol{\xi}'}\left\{\boldsymbol{\xi}' \cdot \boldsymbol{\xi}^* - G\left(\boldsymbol{\xi}'\right)\right\} \tag{20}$$

Then the Hessian of $G^*\left(\boldsymbol{\xi}^*\right)$ is written as

$$\nabla\nabla G^*\left(\boldsymbol{\xi}^*\right) = \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\xi}^*} \tag{21}$$

which is the inverse of the Hessian of $G(\boldsymbol{\xi})$

$$\frac{\partial \boldsymbol{\xi}^*}{\partial \boldsymbol{\xi}} = \nabla \nabla G(\boldsymbol{\xi})$$

The last step is guaranteed by $\nabla G^* = \nabla G^{-1}$, which can be shown from Appendix E.

G Details of Assumptions

Previous investigations into the mirror Langevin algorithm [59] have required the relative μ-strong convexity of f with respect to G to guarantee convergence. However, our work introduces Assumption 4.1, which relaxes this requirement and permits consideration of non-strongly convex distributions. Assumption 4.1 can be transformed as following. Taking $h(x) = \sqrt{\frac{d\rho(x)}{d\pi(x)}}$, then $\forall \rho$

$$\mathbb{D}_{KL}(\rho\|\pi) := \int \rho(x) \log \frac{\rho(x)}{\pi(x)} dx \leq \frac{1}{2\beta} \int \rho(x) \left\| \nabla \log \frac{\rho(x)}{\pi(x)} \right\|^2_{[\nabla^2 G(x)]^{-1}} dx =: \frac{1}{2\beta} J_\pi^G(\rho) \quad (22)$$

The $\mathbb{D}_{KL}(\rho\|\pi)$ term represents the KL divergence, often serving as a measure of the distance between distributions ρ and π. On the other hand, the right-hand side term, $J_\pi^G(\rho)$, signifies the weighted Fisher information. As demonstrated by [22], Langevin dynamics can be interpreted as the gradient flow of the KL divergence within the space of probability measures, equipped with the Wasserstein metric through the Fokker-Planck equation. This connection establishes a link between sampling and optimization.In this context, Assumption 4.1 can be perceived as the condition of gradient domination for KL-divergence in the Wasserstein metric.

Assumption 4.2 specifically relates to the interplay between the higher-order derivatives and the lower-order derivatives of the function. When the secondary derivative is small, it implies that the first derivative, which is governed by the secondary derivative, is also small. This property ensures the solution of continuous dynamics and Hessian stability [59], indicating that the underlying geometry does not undergo rapid changes. Moreover, this property is preserved under Fenchel conjugation (with the same parameter), affine transformation and summation [36]. The concept of self-concordance is also prevalent in quadratic optimizations, such as the interior point method, where it guarantees the convergence performance $O\left(\sqrt{\zeta} \log \frac{1}{\epsilon}\right)$.

In Assumption 4.3, when $G(x) = \frac{||x||^2}{2}$, we regain the conventional definition of a differentiable function being Lipschitz continuous with a parameter β. This property has been extensively employed in prior research. In the case where $G = f$ and a function f satisfies $\|\nabla f(x)\|_{[\nabla^2 f(x)]^{-1}} \leq L$, it is referred to as a barrier function [35]. This property also emerges in the analysis of Newton's method in quadratic optimization scenarios.

In formal algorithms, it is often necessary to have the γ-relative smooth property in order to ensure convergence. γ-relative smooth is defined by

$$\begin{aligned} &\left\| \left[\nabla^2 G(x)\right]^{-1} \nabla f(x) - \left[\nabla^2 G\left(x'\right)\right]^{-1} \nabla f\left(x'\right) \right\|_{\nabla^2 G(x')} \\ &\leq \gamma \cdot \left\| \nabla G(x) - \nabla G\left(x'\right) \right\|_{[\nabla^2 G(x')]^{-1}} \end{aligned} \quad (23)$$

However, the CPMLA utilizes a distinct approach by employing deterministic gradient steps and stochastic steps separately. This allows for the utilization of a weaker notion of smoothness assumption, namely Assumption 4.4. Unlike the γ-relative smooth, which necessitated Lipschitz continuity across different metrics $\nabla^2 G$ and could be unavoidable when discretizing the geometry, this definition of relative smoothness only considers the local metric $\nabla^2 G$ at a single point.

H Proof of Theorem 4.5

Proof. We first clarify which parts of this proof are novel contributions and which are standard techniques adapted from prior work. Our primary contribution is the adaptation of the convergence proof of Mirror Langevin Dynamics to a setting where the mirror map is a learnable neural network

(CP-Flow) and is trained jointly with the target distribution (EBM). The overall structure of the proof, including the use of the Fokker-Planck equation and the mirror LSI, follows the framework established by [21]. Our novel steps include explicitly accounting for the approximation errors from both the EBM and CP-Flow (δ_2 and δ_3) and ensuring the proof holds under standard deep learning practices like gradient clipping.

For the proof, we analyze the convergence of the sampling distribution ρ_t to the stationary distribution of the learned EBM, p_{θ^*}. We therefore assume that p_{θ^*} satisfies the β-Mirror LSI (Assumption 4.1), as p_{θ^*} is trained to be a close approximation of the target data distribution $\pi = p_{\text{data}}$.

We decompose the total variation distance using the triangle inequality:

$$d_{TV}(\rho_t, p_{\text{data}}) \leq d_{TV}(\rho_t, p_{\theta^*}) + d_{TV}(p_{\theta^*}, q_{\vartheta^*}) + d_{TV}(q_{\vartheta^*}, p_{\text{data}}). \tag{24}$$

where p_{θ^*} is the stationary distribution of the energy model and q_{ϑ^*} is the optimal CP-Flow distribution.

For the first term, following Lemma 1 in [21], we analyze the differential form in primal space. Lemma 1 in [21] tells us that the differential form of Algorithm 2 in primal space is

$$\begin{aligned}
dX_t =& -\left[\nabla^2 G\left(X_t\right)\right]^{-1} \nabla f\left(X_0\right) dt - \left[\nabla^2 G\left(X_t\right)\right]^{-1} \operatorname{Tr}\left(\nabla^3 G\left(X_t\right)\left[\nabla^2 G\left(X_t\right)\right]^{-1}\right) dt \\
&+ \sqrt{2\left[\nabla^2 G\left(X_t\right)\right]^{-1}} dW_t \\
=& \Big[-\left[\nabla^2 G(x_t)\right]^{-1} \nabla f(x_0) + \left[\nabla^2 G(x_t)\right]^{-1} \nabla f(x_t) - \left[\nabla^2 G(x_t)\right]^{-1} \nabla f(x_t) \\
&- \left[\nabla^2 G\left(X_t\right)\right]^{-1} \operatorname{Tr}\left(\nabla^3 G\left(X_t\right)\left[\nabla^2 G\left(X_t\right)\right]^{-1}\right)\Big] dt + \sqrt{2\left[\nabla^2 G\left(X_t\right)\right]^{-1}} dW_t \\
=& \left(\nabla \cdot H^{-1}\left(X_t\right) - H^{-1}\left(X_t\right) \nabla f\left(X_t\right) + \hat{\mu}\right) dt + \sqrt{2H^{-1}\left(X_t\right)} dW_t
\end{aligned} \tag{25}$$

where we denote $\hat{\mu} = \left[\nabla^2 G\left(X_t\right)\right]^{-1}\left(\nabla f\left(X_t\right) - \nabla f\left(X_0\right)\right)$ and $H^{-1} = [\nabla^2 G]^{-1}$.

This is a weighted Langevin dynamics with shifted drift $\hat{\mu}$ (the reason of the convergence to a biased limit).

Now consider the Fokker-Planck equation for the conditional density $\rho_{t|0}\left(x_t \mid x_0\right)$. For the drift $b = \nabla \cdot H^{-1} - H^{-1}\nabla f + \hat{\mu}$, applying Lemma 3 in [51], we have

$$\begin{aligned}
\frac{\partial \rho_t(x)}{\partial t} &= \int \frac{\partial \rho_{t|0}\left(x \mid x_0\right)}{\partial t} \rho_0\left(x_0\right) dx_0 \\
&= \int \left[-\nabla \cdot \left(\rho_{t|0}\left(\nabla \cdot G_0(x) - G_0(x)\nabla f(x)\right)\right) + \left\langle \nabla^2, \rho_{t|0} G_0(x)\right\rangle - \nabla \cdot \left(\rho_{t|0}\hat{\mu}_0(x)\right)\right] \rho_0\left(x_0\right) dx_0 \\
&= \nabla \cdot \left(\rho_{0|t} \int -\left(\rho_t\left(\nabla \cdot G_0(x) - G_0(x)\nabla f(x)\right)\right) + \nabla \cdot \left(\rho_t G_0(x)\right) dx_0\right) - \nabla \cdot \left(\rho_t \int \rho_{0|t}\hat{\mu}_0(x) dx_0\right) \\
&= \nabla \cdot \left(\rho_{0|t} \int \rho_t G_0 \nabla \log \frac{\rho_t}{p_{\theta^*}(x)} dx_0\right) - \nabla \cdot \left(\rho_t \int \rho_{0|t}\hat{\mu}_0(x) dx_0\right) \\
&= \nabla \cdot \left(\rho_{0|t} \int \rho_t G_0 \nabla \log \frac{\rho_t}{p_{\theta^*}(x)} dx_0\right) - \nabla \cdot \left(\rho_t \int \rho_{0|t}\hat{\mu}_0(x) dx_0\right)
\end{aligned} \tag{26}$$

where the last equality is because $\nabla \log \frac{\rho}{p_{\theta^*}} = \nabla(\log \rho + f_{\theta^*})$.

Now consider the KL-divergence

$$\frac{d}{dt}\mathbb{D}_{KL}(\rho_t \| p_{\theta^*}) = \int \frac{d\rho_t}{dt} \log \frac{\rho_t}{p_{\theta^*}} dx + \int p_{\theta^*} \frac{1}{p_{\theta^*}} \frac{d\rho_t}{dt} dx = \int \frac{d\rho_t}{dt} \log \frac{\rho_t}{p_{\theta^*}} dx \tag{27}$$

According to Equation 26, we have

$$\begin{aligned}
\frac{d}{dt}\mathbb{D}_{KL}(\rho_t\|p_{\theta^*}) &= \int \frac{d\rho_t}{dt}\log\frac{\rho_t}{p_{\theta^*}}dx \\
&= \int \nabla\cdot\left(\rho_{0|t}\int \rho_t G_0 \nabla\log\frac{\rho_t}{p_{\theta^*}}dx_0\right)\log\frac{\rho_t}{p_{\theta^*}}dx - \int\nabla\cdot\left(\rho_t\int\rho_{0|t}\hat{\mu}_0 dx_0\right)\log\frac{\rho_t}{p_{\theta^*}}dx \\
&= -\int\rho_{0|t}\int\rho_t\left\langle\nabla\log\frac{\rho_t}{p_{\theta^*}}G_0,\nabla\log\frac{\rho_t}{p_{\theta^*}}\right\rangle dx_0dx + \int\rho_t\int\rho_{0|t}\left\langle\hat{\mu},\nabla\log\frac{\rho_t}{p_{\theta^*}}\right\rangle dx_0dx \\
&= -\mathbb{E}_{\rho_t}\left[\left\|\nabla\log\frac{\rho_t}{p_{\theta^*}}\right\|^2_{[\nabla^2 G]^{-1}}\right] + \mathbb{E}_{\rho_{0,t}}\left[\left\langle\hat{\mu},\nabla\log\frac{\rho_t}{p_{\theta^*}}\right\rangle\right] \\
&\le -\mathbb{E}_{\rho_t}\left[\left\|\nabla\log\frac{\rho_t}{p_{\theta^*}}\right\|^2_{[\nabla^2 G]^{-1}}\right] + \mathbb{E}_{\rho_{0,t}}\left[\|\hat{\mu}\|^2_{\nabla^2 G}\right] + \frac{1}{4}\mathbb{E}_{\rho_t}\left[\left\|\nabla\log\frac{\rho_t}{p_{\theta^*}}\right\|^2_{[\nabla^2 G]^{-1}}\right] \\
&\le -\frac{3\beta}{2}\mathbb{D}_{KL}(\rho_t\|p_{\theta^*}) + \mathbb{E}_{\rho_{0,t}}\left[\|\hat{\mu}\|^2_{\nabla^2 G}\right]
\end{aligned} \tag{28}$$

The third equality refers to the integration by parts formula $\int\langle\nabla G(x), v(x)\rangle dx = -\int G(x)\nabla\cdot v(x)dx$. The first inequality is because $x^\top y \le \|x\|_2^2 + \frac{1}{4}\|y\|_2^2$ and the last inequality is from Mirror LSI (Assumption 4.1).

Under Assumption 4.2 - 4.4, let $M := \exp\left(2\zeta D/\sqrt{\alpha}\right)$. We have

$$\begin{aligned}
\mathbb{E}_{\rho_{0,t}}\left[\|\hat{\mu}\|^2_{\nabla^2 G}\right] &\le \gamma^2\cdot\mathbb{E}_{\rho_{0,t}}\left[\left\|\nabla G\left(x_t\right) - \nabla G\left(x_0\right)\right\|^2_{[\nabla^2 G(x_t)]^{-1}}\right] \\
&= \gamma^2\cdot\mathbb{E}\left[\left\|-t\nabla f\left(x_0\right) + \sqrt{2}\int_0^t\left[\nabla^2 G\left(x_s\right)\right]^{1/2}dW_s\right\|^2_{[\nabla^2 G(x_t)]^{-1}}\right] \\
&\le 2\gamma^2 t^2\mathbb{E}\left\|\nabla f\left(x_0\right)\right\|^2_{[\nabla^2 G(x_t)]^{-1}} + 4\mathbb{E}\int_0^t\left\|\nabla^2 G\left(x_s\right)\right\|_{[\nabla^2 G(x_t)]^{-1}}ds \\
&\le 2\gamma^2 t^2 L^2 + 4t\gamma^2 M d
\end{aligned} \tag{29}$$

where the second inequality we use Itô isometry and $(a+b)^2 \le 2(a^2+b^2)$.

Then if $0 \le t \le h$, we have

$$\frac{d}{dt}\mathbb{D}_{KL}(\rho_t\|p_{\theta^*}) \le -\frac{3\beta}{2}\mathbb{D}_{KL}(\rho_t\|p_{\theta^*}) + 2\gamma^2 h^2 L^2 + 4h\gamma^2 M d \tag{30}$$

which is

$$\frac{d}{dt}\left(e^{\frac{3\beta}{2}t}\mathbb{D}_{KL}(\rho_t\|p_{\theta^*})\right) \le e^{\frac{3\beta}{2}t}\left(2\gamma^2 h^2 L^2 + 4h\gamma^2 M d\right) \tag{31}$$

Integrate it for $0 \le t \le h$,

$$e^{\frac{3\beta}{2}h}\mathbb{D}_{KL}(\rho_h\|p_{\theta^*}) - \mathbb{D}_{KL}(\rho_0\|p_{\theta^*}) \le \frac{2}{3\beta}\left(e^{\frac{3\beta h}{2}} - 1\right)\left(2\gamma^2 h^2 L^2 + 4h\gamma^2 M d\right) \tag{32}$$

Then

$$\mathbb{D}_{KL}(\rho_h\|p_{\theta^*}) \le e^{-\frac{3\beta}{2}h}\mathbb{D}_{KL}(\rho_0\|p_{\theta^*}) + \frac{2}{3\beta}(1 - e^{-\frac{3\beta h}{2}})\left(2\gamma^2 h^2 L^2 + 4h\gamma^2 M d\right) \tag{33}$$

Iterating the recursion,

$$\mathbb{D}_{KL}(\rho_k\|p_{\theta^*}) \le e^{-\frac{3\beta}{2}hk}\mathbb{D}_{KL}(\rho_0\|p_{\theta^*}) + \frac{2}{3\beta}\left(2\gamma^2 h^2 L^2 + 4h\gamma^2 M d\right) \tag{34}$$

Using Lemma 6 in [21] for initialization, picking the assumed stepsize, after $k \ge \tilde{\Omega}\left(M\gamma^2 d/\beta^2\delta\right)$, we have $\mathbb{D}_{KL}(\rho_t\|p_{\theta^*}) < \delta$.

Using Pinsker's inequality, we establish:

$$d_{TV}(\rho_t, p_{\theta^*}) \leq \sqrt{\frac{1}{2} D_{KL}(\rho_t | p_{\theta^*})} < \sqrt{\frac{\delta_1}{2}} \tag{35}$$

The second term, $d_{TV}(p_{\theta^*}, q_{\vartheta^*})$, represents the approximation error from running the LMC for a finite number of steps T instead of running it to convergence. This term can be bounded by summing the incremental changes over the T steps. Let $p_0 = p_{\theta^*}$ and $p_T = q_{\vartheta^*}$ be the distributions at the start and end of the sampling chain. By the triangle inequality for TV distance, we have $d_{TV}(p_{\theta^*}, q_{\vartheta^*}) \leq \sum_{t=1}^{T} d_{TV}(p_t, p_{t-1})$. We can analyze the single-step change using the Fokker-Planck equation:

$$\frac{\partial p_t(x)}{\partial t} = -\nabla_x \cdot \left(p_t(x) \frac{\eta^2}{2} \nabla_x f_\theta(x) \right) + \frac{\eta^2}{2} \nabla_x^2 p_t(x) \tag{36}$$

From this, we can estimate the incremental change as $d_{TV}(p_t, p_{t-1}) \leq \sqrt{\frac{1}{2} D_{KL}(p_t | p_{t-1})} \sim O(\eta)$. Summing over T steps gives a total error of: $d_{TV}(p_{\theta^*}, q_{\vartheta^*}) \sim O(T\eta)$.

The third term leverages the universality property of CP-Flow (Theorem 3 in [20]). Given that the initial noise distribution is absolutely continuous with respect to the Lebesgue measure, there exists a sequence q_{ϑ_n} such that $d_{TV}(q_{\vartheta_n}, p_{\text{data}}) < \delta_3$ as $n > N$.The optimality of CP-Flow (Theorem 4 in [20]) further guarantees almost sure convergence in distribution of q_{ϑ_n} to the optimal Brenier map q_{ϑ^*}, ensuring that $d_{TV}(q_{\vartheta^*}, p_{\text{data}}) < \delta_3$.

Combining these results, we conclude that $d_{TV}(\rho_t, p_{\text{data}}) < \delta = \sqrt{\frac{\delta_1}{2}} + \delta_2 + \delta_3$.

□

I Experimental Details

Parameter	Value
Dataset Size	50,000 samples
Dynamic Mirror Map ∇G	1 CP-Flow block
Depth	20
Dimh	32
∇G Optimizer	Adam
∇G Activation	Gaussian Softplus
∇G Initial Learning Rate	0.005
EBM ∇f	4 linear layers
∇f Optimizer	Adam
∇f Activation	Swish
∇f Initial Learning Rate	0.005
Batch Size	128
Reported Results	After 3 and 10 epochs

Table 4: Experimental setup for toy dataset

Table 4 outlines the experimental setup for an Eight Gaussian toy dataset experiment. This setup includes a dataset size of 50,000 samples, using single CP-Flow block with the Gaussian Softplus activation function for the Dynamic Mirror Map ∇G. The optimizer for ∇G is Adam, with a and an initial learning rate of 0.005. The EBM ∇f comprises four linear layers with Swish activation, also utilizing the Adam optimizer, and an initial learning rate of 0.005. The batch size for this setup is 128, with reported results after 3 and 10 epochs.

Table 5 presents the experimental setup for the CIFAR-10, SVHN, and CelebA datasets. We use a multi-scale structure, involving two CP-Flow blocks, followed by invertible downsampling, and then another two CP-Flow blocks. All ICNN architectures had two hidden layers. The strong convexity

Parameter	Value
Datasets	CIFAR-10, SVHN, CelebA
Dynamic Mirror Map ∇G	2 CP-Flow blocks
ICNN Architecture	2 hidden layers
∇G Optimizer	Adam
∇G Activation	Gaussian Softplus
∇G Initial Learning Rate	5e-4
∇G Weight Decay	5e-5
Mirror Steps	10
Mirror Step Size	1e-2
EBM ∇f	3 blocks with 3 convolutional layers
∇f Optimizer	Adam
∇f Activation	Swish
∇f Initial Learning Rate	5e-3
Batch Size	128
Reported Results	After 200 epochs

Table 5: Experimental setup for CIFAR-10, SVHN, and CelebA datasets

	CIFAR-10	SVHN	CelebA
Training time (hours)	10.5	12.6	40.2

Table 6: Training time on each dataset on eight 3090 GPUs

parameter α (Section 2.2) is set to 1e-4. For the EBM in CPMLA, a 3 blocks network is used to design the negative energy function. The following Table 6 presents training time of our model on each dataset on eight 3090 GPUs.

In the CPMLAprt setting, we first pretrain a CP-Flow on training examples, and then train a 30-step mirror Langevin sampling, whose parameters are initialized randomly, together with the pretrained CP-Flow by following Algorithm 2.

J More Image Generation Results

In Section 5.2, we have shown generated examples from CPMLA. In this section, we first show the examples generated by LMC on CIFAR-10. Then we compare the examples generated by the CP-Flow only and CPMLAprt in Figure 8 and 9. We can see huge difference between two algorithms and our generated examples are meaningful.

Models	**#Para**	**FID↓**
NT-EBM	23.8M	78.12
EBM-FCE	44.9M	37.30
GLOW	44.2M	45.99
Flow++ only	28.8M	92.10
CoopFlow	45.9M	21.16
CPMLA	17.39M	20.85

Table 7: Model size vs. performance comparison (lower is better). CPMLA is lightweight yet highly effective, showcasing superior efficiency.

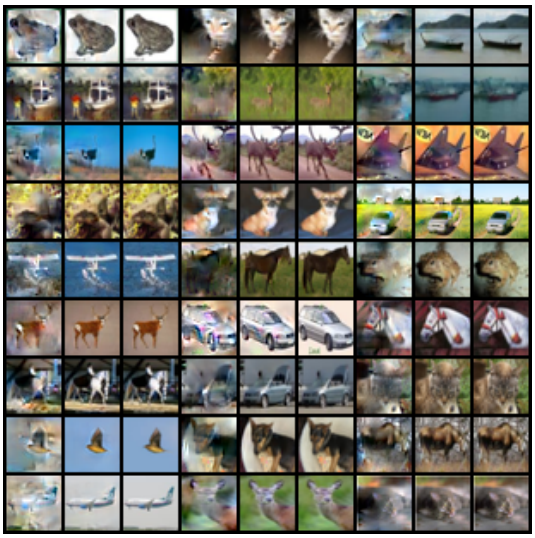


Figure 6: Image reconstruction on the CIFAR-10. The right column showcases the original images. The left and middle columns feature flow-generated images and the reconstructed images, respectively. We can see that the reconstruction is almost the same as the original one, which solidifies the stance that CPMLA functions effectively as a sampling algorithm.

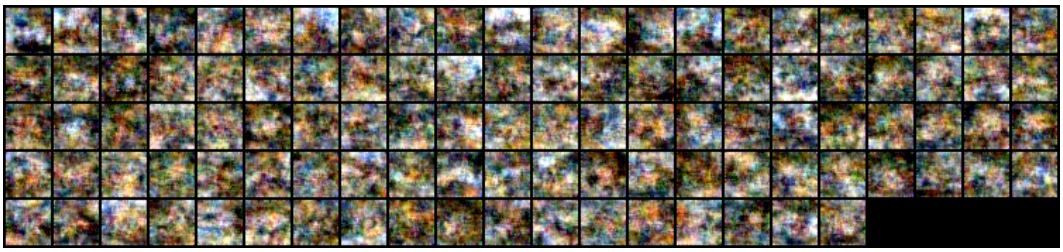
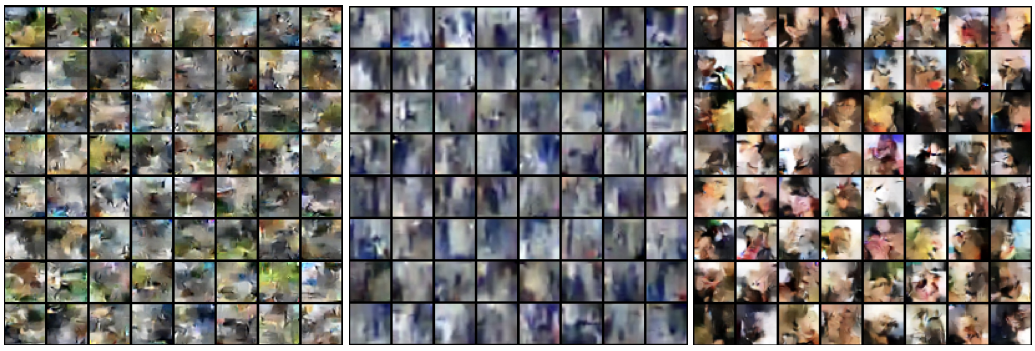
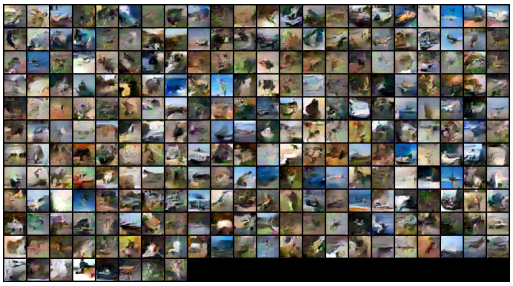


Figure 7: LMC on CIFAR-10



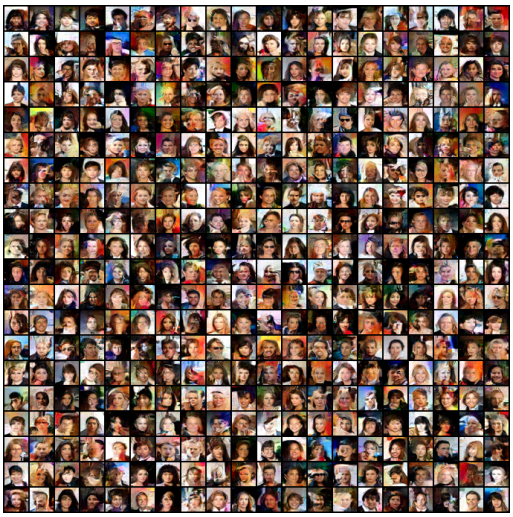
(a) CIFAR-10 (b) SVHN (c) CelebA

Figure 8: CP-Flow results



(a) CPMLA CIFAR-10

(b) CPMLAprt CIFAR-10



(c) CPMLAprt CelebA

(d) CPMLAprt SVHN

Figure 9: CPMLAprt results