FOLLOW HAMILTONIAN LEADER: AN EFFICIENT ENERGY-GUIDED SAMPLING METHOD

Yunfei Teng[1] **Sixin Zhang**[2] **Yao Li**[1] **Kai Chen**[1] **Di He**[3] **Qiwei Ye**[1]
[1]Beijing Academy of Artificial Intelligence (BAAI), China
[2]IRIT, INP, Univ. Toulouse, France
[3]Peking University (PKU), China

ABSTRACT

Our research underscores the value of leveraging *zeroth-order* information for addressing sampling challenges, particularly when first-order information is unreliable or unavailable. In light of this, we have developed a novel parallel sampling method that incorporates a leader-guiding mechanism that preserves the same invariant measure property as Hamiltonian Monte Carlo. This mechanism forges connections between multiple sampling instances via a determined leader, enhancing both the efficiency and effectiveness of the entire sampling process. Our experimental results demonstrate that our method markedly expedites the exploration of the target distribution and produces superior quality outcomes compared to traditional sampling techniques. Furthermore, our method also shows greater resilience against the detrimental impacts of corrupted gradients as intended.

1 INTRODUCTION

Score-based generative models Sohl-Dickstein et al. (2015); Song & Ermon (2019); Ho et al. (2020) introduce a novel approach to generative modeling that revolves around the estimation and sampling of the Stein score Liu et al. (2016); Song & Ermon (2019). The score represents the gradient of the log-density function $\nabla_x \log \pi(x)$ evaluated at the input data point x. This type of approach usually relies on effectively training a deep neural network to accurately estimate the score. The estimated score is then utilized to navigate the sampling process, ultimately resulting in the production of high-quality data samples that closely match the areas of high density in the original distribution.

In our research, we investigate the sampling of a probability distribution given by $\pi(x) \propto e^{-U(x)}$, where $U(x)$ is the energy function. In the context of energy-based score-matching generative models, the objective often involves sampling the modes in areas of high probability density. An approach as suggested in Song & Ermon (2019); Ho et al. (2020), is to smooth the original distribution by convolving $\pi(x)$ with an isotropic Gaussian distribution of variance σ^2, yielding $\pi_\sigma(x) = \int \pi(x')\mathcal{N}(x; x', \sigma^2 I)\, dx'$. By gradually decreasing the variance σ^2, $\pi_\sigma(x)$ recovers the original distribution $\pi(x)$.

Typically, the sampling of score-based approaches are integrated with numerical SDE solvers Song et al. (2021), for example, the Euler-Maruyama solver, as well as Monte Carlo Markov Chain (MCMC) techniques like Langevin Dynamics Parisi (1981). Furthermore, there is a notable similarity between score-based sampling algorithms and first-order optimization algorithms. Efforts have been made to merge these two methodologies, particularly from a perspective of sampling Welling & Teh (2011); Chen et al. (2014b; 2016). All these methods primarily concentrates on first-order information $\nabla_x U(x)$ to improve performance, while typically treating the zeroth-order information $U(x)$ merely as a basis for rejecting samples Hastings (1970); Roberts & Tweedie (1996); Neal (2011).

We argue that incorporating zeroth-order information can significantly enhance the algorithm's overall effectiveness, particularly in instances where the first-order information is compromised. To address this, we draw inspiration from parallel tempering Swendsen & Wang (1986), a simulation method commonly used to identify the lowest energy state in systems of interacting particles. The fundamental principle of parallel tempering involves operating multiple sampling replicas simultaneously, each at a different temperature level. These temperatures typically range from low, where the system is prone to being trapped in local minima, to high, which facilitates the system's ability to surmount energy barriers and more thoroughly explore the energy landscape.

Drawing inspiration from this concept, we extend the Hamiltonian Monte Carlo (HMC) framework Neal (2011) and introduce a novel algorithm that concurrently runs multiple replicas, sampling at both high and low Hamiltonian energy levels. Moreover, this methodology combines both zeroth and first order information from various chains, hence enhancing the effectiveness of sampling approaches. The experimental findings demonstrate the efficacy of our approach in scenarios where relying solely on first-order knowledge is insufficient. These findings illustrate the capacity of incorporating zeroth-order information to greatly enhance the efficiency and accuracy of sampling operations in energy-based score-matching algorithms.

2 Background

2.1 Hamiltonian Monte Carlo

The primary purpose of MCMC is to construct a Markov chain that matches its equilibrium distribution to the target distribution. One of the most popular MCMC methods is Langevin Monte Carlo Grenander & Miller (1994); Roberts & Tweedie (1996), which proposes samples in a Metropolis-Hastings Hastings (1970) framework for more efficient state space exploration. Another advanced method is HMC Neal (2011); Chen et al. (2014a); Betancourt (2018), which incorporates an auxiliary variable p and employs Hamiltonian dynamics to facilitate the sampling process. The Hamiltonian function is structured as a composite of potential energy $U(x)$ and kinetic energy $K(p)$, defined as:

$$H(x,p) = U(x) + K(p), \tag{1}$$

where x represents the position of a particle and p denotes its momentum. Kinetic energy $K(p)$ is commonly formulated as $K(p) = \frac{1}{2}p^T M^{-1} p$, where M corresponds to the mass matrix. For simplicity, we assume in this paper that the mass matrix M is equal to the identity matrix I. The joint distribution of position and momentum conforms to the canonical distribution:

$$\pi(x,p) = e^{-H(x,p)}/Z, \tag{2}$$

where $Z = \iint e^{-H(x,p)}\, dxdp$. Samples from $\pi(x)$ can then be obtained by marginalizing p from $\pi(x,p)$, which further requires $\int_p \pi(x,p)\, dp =$ constant. In the HMC algorithm, proposals are generated by simulating Hamiltonian dynamics and then subjected to a Metropolis criterion to determine their acceptance or rejection. A commonly employed numerical method for solving these equations is the *Leapfrog* integrator Birdsall & Langdon (2005).

Recent progress in HMC techniques has focused on increasing their adaptability and applicability in a variety of contexts. Such developments include the NUTS sampler Hoffman & Gelman (2011), which features an automatic mechanism for adjusting the number of simulation steps. The Riemann manifold HMC Girolami et al. leverages Riemannian geometry to modify the mass matrix M, making use of curvature information to improve sampling efficiency. Stochastic Gradient Hamiltonian Monte Carlo Chen et al. (2014a); Ma et al. (2015) investigates a stochastic gradient approach within the HMC framework. Our contribution is distinct from these methods and can be integrated with them.

2.2 Energy-based score-matching model

Probabilistic models often require normalization, which can become infeasible when dealing with high-dimensional data LeCun et al. (2006); Du & Mordatch (2019). Since the exact probabilities of less probable alternatives become less crucial as long as they remain relatively lower, rather than solely predicting the most probable outcome, models can be structured to interpret relationships between variables via an energy function. Within the context of generative models, these energy-based models (EBMs) are devised to assign higher energy values to regions of lower probability and lower energy values to regions of higher probability.

Score matching Hyvärinen (2005); Song & Ermon (2019) is a statistical estimation technique that learns probability distributions by directly modeling the score function (the gradient of the log-density) rather than the density itself. This approach circumvents the need for explicit normalization in high-dimensional spaces, where traditional density estimation becomes computationally intractable. The method works by minimizing the discrepancy between the model's score function and the empirical scores derived from observed data, enabling efficient estimation of complex distributions without requiring normalized probability densities.

3 Motivation

In our work, we assume to have access to both the gradient information $\nabla_x U(x)$ as well as the energy information $U(x)$. In certain scenarios, gradients may yield information that is either of limited or potentially detrimental value value. Our research examines situations where gradients are compromised, highlighting the importance of zeroth-order information, often associated with energy-based sampling.

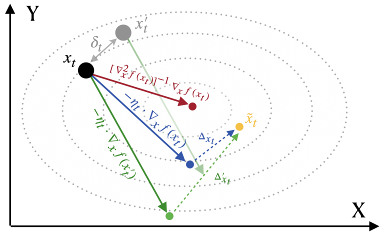

Y
X
x'_t
δ_t
x_t
$[\nabla_x^2 f(x_t)]^{-1} \nabla_x f(x_t)$
$-\eta_t \cdot \nabla_x f(x_t)$
$-\eta_t \cdot \nabla_x f(x'_t)$
$\tilde{x}_t$
Δx_t
$\Delta' x_t$

Figure 1: A good *anchor point* could help improve convergence even if the gradient is *unexpectedly disturbed* from *original gradient* to the *disturbed gradient*, getting closer to the optimal point.

In high-dimensional spaces, sampling algorithms may encounter difficulties in converging when faced with a complex probability distribution. This instability can occur when the local Hessian matrix is ill-conditioned or when the spectrum of the local Hessian matrix is extremely large. These circumstances frequently give rise to inaccuracies or instabilities in numerical computations, which might cause the convergence process to break down. The samples produced may deviate significantly from the true mode, leading to poor sample quality.

Nevertheless, as shown in Figure 1, using an anchor point can boost the stability of convergence. Additionally, when particles tend to become trapped in local minima due to uninformative gradients (for instance, at a saddle point or on a flat loss landscape), this method can enhance performance.

In addition, certain situations may present a divergence between the gradient information and the ground truth. This divergence can hinder algorithms from accurately converging to the appropriate modes. In these instances, it becomes essential to incorporate energy information to rectify inaccuracies that arise from solely depending on gradients.

4 Algorithm

Many sampling methods typically rely on independent Markov chains, which can lead to the issues mentioned in Section 3. Taking inspiration from Swendsen & Wang (1986), our approach involves the utilization of multiple replicas. This approach enables us to implicitly encourage greater exploration among multiple particles while simultaneously preserving the optimal outcomes for exploitation purposes. We will elaborate on how our algorithm can be employed to tackle these challenges.

Algorithm 1 Elastic Leapfrog (eLeapfrog)

```
Input: A collection of positions $\{x^i\}_{i=1}^n \in \mathbb{R}^{n\times d}$, a collection of momenta $\{p^i\}_{i=1}^n \in \mathbb{R}^{n\times d}$, learning rate $\eta > 0$, pulling strength $\lambda \geq 0$, number of Leapfrog steps $L$.
for $k = 1, \cdots, L$ do
    $x^l \leftarrow \text{Leader}\left(\{x^i\}_{i=1}^n\right)$
    for $i = 1, \cdots, n$ do $p^i \leftarrow p^i - \frac{\eta}{2} \cdot \nabla_x U_e(x^i; x^l)$ end for
    for $i = 1, \cdots, n$ do $x^i \leftarrow x^i + \eta \cdot p^i$ end for
    $x^l \leftarrow \text{Leader}\left(\{x^i\}_{i=1}^n\right)$
    for $i = 1, \cdots, n$ do $p^i \leftarrow p^i - \frac{\eta}{2} \cdot \nabla_x U_e(x^i; x^l)$ end for
end for
Output: $\{x^i\}_{i=1}^n \in \mathbb{R}^{n\times d}, \{p^i\}_{i=1}^n \in \mathbb{R}^{n\times d}$
```

First, we present a modified version of the leapfrog method, termed the *Elastic Leapfrog* (eLeapfrog). In this approach, extra elastic forces are applied between each particle and a designated *leader*, effectively incorporating an additional elastic energy term into the standard Hamiltonian. This modification augments the standard Hamiltonian with an additional elastic energy term, effectively preventing particle divergence and promoting local exploitation. Next, we introduce a *leader-pulling* mechanism designed to significantly boost the particles' exploratory capabilities. Finally, combining these innovations, we develop the Follow Hamiltonian Leader (FHL) algorithm, which synergistically integrates first-order and zeroth-order information to achieve superior sampling efficiency compared to conventional approaches.

4.1 Elastic Leapfrog

To improve the efficiency of sampling, we integrate an elastic force component into the conventional leapfrog technique. This enhancement aims to dynamically guide particles towards a leading particle, facilitating their movement and improve their exploration ability. The method could be treated like temporarily storing potential energy within an elastic spring, which is then converted into kinetic energy. By adding extra elastic force, we could define the energy of elastic HMC as:

$$H_e(x, p; x^l) = U_e(x; x^l) + K(p), \quad \text{where} \quad U_e(x; x^l) = U(x) + \frac{\lambda}{2}\|x - x^l\|_2^2 \tag{3}$$

In practice, the leader x^l is selected based on the energy levels of the workers. In other words, we aim to choose a leader that is close to the worker with the lowest energy (i.e., the highest probability). At each iteration, the leader x^l is computed as the weighted average

$$x^l = \text{Leader}(\{x^i\}_{i=1}^n) = \frac{\sum_{i=1}^n \exp\left(-\beta\, f(x^i)\right) \cdot x^i}{\sum_{i=1}^n \exp\left(-\beta\, f(x^i)\right)}, \tag{4}$$

where β is a scaling factor. In our experiment, we found that $\beta = 1$ works well acrooss the experiments. By integrating the leader selection technique with the conventional leapfrog method, we obtain the eLeapfrog algorithm, as described in Algorithm 1.

4.2 Leader Pulling

Next, we introduce our *leader pulling* method. In this approach, we define a transition kernel such that for any given particle x, a new position x' is sampled from

$$q(x' \mid x,\, x^l) = \mathcal{N}\Big(x';\, (1-\gamma)\cdot x + \gamma \cdot x^l,\, \sigma_l^2 I\Big). \tag{5}$$

Subsequently, we employ the Metropolis-Hastings algorithm to determine the acceptance probability for the proposed move:

$$\alpha_{\text{init}} = \min\left\{1,\ \prod_{i=1}^{n} \frac{e^{-U(x^i_{\text{init}})}\, q(x^i \mid x^i_{\text{init}}, x^l_{\text{init}})}{e^{-U(x^i)}\, q(x^i_{\text{init}} \mid x^i, x^l)}\right\},$$

where x^l and x^l_{init} are determined as described in Equation (4). This method substantially enhances the exploration capabilities of the algorithm, echoing the approach in (Chen et al., 2024), where incorporating additional noise is shown to significantly improve exploration.

4.3 Follow Hamiltonian Leader

Incorporating zeroth-order information (i.e., function values rather than derivatives) serves two key purposes. Firstly, it provides a search direction that accelerates convergence and helps mitigate issues arising from corrupted first-order information (i.e., gradient inaccuracies), thereby speeding up the optimization process. Second, it helps ensure that we are sampling from the correct underlying distribution by properly accepting or rejecting the proposal.

To ensure that the sampling method maintains detailed balance—a requirement for most sampling algorithms—we evaluate the joint distribution of a group of particles. This evaluation determines whether to accept or reject a proposed move for the whole group, thereby preserving the integrity of the sampling process. This adaptation results in the creation of our algorithm FHL, extensively elucidated in Algorithm 2.

Since we continuously resample the momentum and employ a leader pulling scheme with a Gaussian noise added to the position variable, the chain generated by FHL by construction to be irreducible, aperiodic, and Harris recurrent. Moreover, we will demonstrate that our method also satisfies detailed balance with respect to the target distribution.

Algorithm 2 Follow Hamiltonian Leader

```
Input: A collection of initial positions $\{x_0^i\}_{i=1}^n \in \mathbb{R}^{n\times d}$, learning rate $\eta > 0$, pulling strength $\lambda \geq 0$, number of steps $L$.

for $t = 1, 2, \cdots, T$ do
    # 1. Elastic Leapfrog (refer to Section 4.1)
    for $i = 1, \cdots, n$ do
        Randomly sample the momentum $p_{t-1}^i \sim \mathcal{N}(0, I)$
    end for
    $\{x_{\text{prop}}^i\}_{i=1}^n, \{p_{\text{prop}}^i\}_{i=1}^n \leftarrow$ eLeapfrog $(\{x_{t-1}^i\}_{i=1}^n, \{p_{t-1}^i\}_{i=1}^n, \eta, \lambda, L)$

    Sample a random variable $u \sim \text{Uniform}(0, 1)$
    if $u < \prod_{i=1}^n \exp\left(H(x_{\text{prop}}^i, p_{\text{prop}}^i) - H(x_{t-1}^i, p_{t-1}^i)\right)$ then        ▷ Metropolis-Hastings step
        for $i = 1, \cdots, n$ do $x_t^i \leftarrow x_{prop}^i,\ p_t^i \leftarrow p_{\text{prop}}^i$ end for
    else
        for $i = 1, \cdots, n$ do $x_t^i \leftarrow x_{t-1}^i,\ p_t^i \leftarrow p_{t-1}^i$ end for
    end if

    # 2. Leader Pulling (refer to Section 4.2)
    Draw $\{x_{init}^i\}_{i=1}^n$ by Equation (5)
    Sample a random variable $u \sim \text{Uniform}(0, 1)$
    if $u < \alpha_{init}$ then        ▷ Metropolis-Hastings step
        for $i = 1, \cdots, n$ do $x_t^i \leftarrow x_{init}^i$ end for
    end if
end for

Output: $X_T = \{x_T^i\}_{i=1}^n \in \mathbb{R}^{n\times d}$
```

Detailed Balance Analysis While the leader pulling mechanism in #2 of Algorithm 2 satisfies detailed balance via the Metropolis-Hastings algorithm, we still need to establish that the elastic leapfrog step also preserves detailed balance. To this end, we extend the classical detailed balance properties of HMC (Neal, 2011) to analyze the proposed algorithm.

Denote the combined state of positions and momenta by

$$S = \mathbb{R}^{n\times d} \times \mathbb{R}^{n\times d}.$$

We define the state of FHL as

$$s = \left(\{x^i\}_{i=1}^n, \{p^i\}_{i=1}^n\right) \in S.$$

The detailed balance property of FHL guarantees that the joint probability density π^n on S is preserved over iterations:

$$\pi^n(s) = \prod_{i=1}^n \pi(x^i, p^i) \propto \prod_{i=1}^n e^{-H(x^i, p^i)}.$$

This formulation implies that, although each particle x^i is coupled with the others through a leader, it can still behave independently. To establish the detailed balance property, we introduce the following assumption.

Assumption 1. The probability density π^n is fully supported on the state space S.

Theorem 1. *Under Assumption 1, FHL (Algorithm 2) preserves the invariance of the density π^n, i.e.*

$$\pi^n(ds') = \int p(ds'|s)\pi^n(s)\, ds.$$

where $p(ds'|s)$ is the transition probability kernel at each iteration $t \in \{1, \cdots, T\}$.

The proof is given in Appendix B. It is based on a classical detailed balance result in Tierney (1998) where the key is to verify the reversibility property of the Elastic Leapfrog step #1 so that FHL can ensure the invariance of π^n.

5 EXPERIMENT

In this section, we showcase the efficacy of incorporating zeroth-order information, specifically energy information, into our proposed method to improve the sampling process. To evaluate our method on the performance of the concerned questions, we conduct a comparative analysis against the following baseline algorithms:

- **LMC (Langevin Monte Carlo)**: An MCMC method as described in Grenander & Miller (1994) that uses Langevin dynamics to sample from probability distributions. It is also known as the Metropolis-adjusted Langevin algorithm.
- **HMC (Hamiltonian Monte Carlo)**: An MCMC algorithm that employs Hamiltonian dynamics for more efficient traversal of the state space, leading to better exploration and sampling from complex distributions Neal (2011); Chen et al. (2014a); Betancourt (2018).
- **U-LMC (Unadjusted Langevin Dynamics)**: A variation of LMC without the Metropolis correction, referred to Roberts & Tweedie (1996); Andrieu et al. (2010); Welling & Teh (2011).
- **U-HMC (Unadjusted Hamiltonian Monte Carlo)**: A form of HMC that excludes the Metropolis correction step, as in Sohl-Dickstein et al. (2014); Geffner & Domke (2021).

5.1 GAUSSIAN MIXTURE MODEL

5.1.1 EXPLORATION

To showcase the exploratory capabilities of FHL, we construct a Gaussian mixture model defined as

$$\pi(x) = \sum_{i=1}^{5} w_i \mathcal{N}(\mu_i, \Sigma)$$

In this model, the weight coefficients w_i and the mean vectors μ_i are specified as

$$\{w_i\}_{i=1}^5 = \left\{ \frac{1}{41}, \frac{4}{41}, \frac{4}{41}, \frac{16}{41}, \frac{16}{41} \right\} \quad \text{and} \quad \{\mu_i\}_{i=1}^5 = \{(0,0),\ (2,0),\ (-2,0),\ (4,0),\ (-4,0)\},$$

respectively. The covariance matrix is given by the diagonal matrix

$$\Sigma = \text{diag}(0.04, 1.0).$$

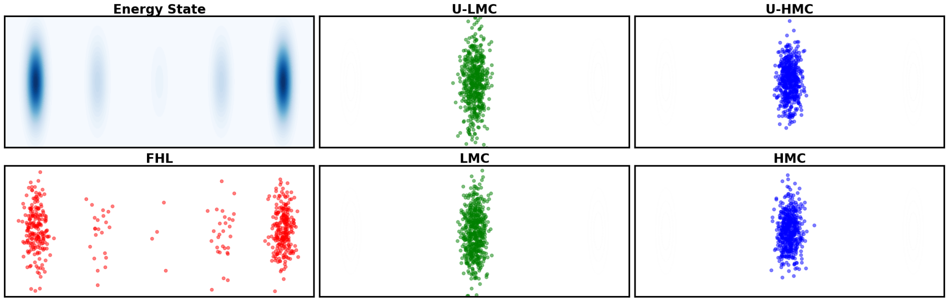


Figure 2: Visualization of $N = 512$ samples X_T drawn from a 5-mode Gaussian mixture model in $d = 2$, starting from the central mode. The plot displays the energy landscape corresponding to the target density π. After $T = 500$ iterations, both the baseline methods (U-LMC, LMC, U-HMC, HMC) and our proposed method (FHL) generate X_T from the same initial set $X_0 = \{x_0^i\}_{i=1}^n$ with each $x_0^i = (0, 0)$. Due to the weakening of the gradient flows between modes, the baseline methods tend to get trapped in metastable states.

In our experiments, we initialize all particles at the origin $(0, 0)$ and then continuously sample from the specified distribution. It is important to note that during transitions between modes, the gradient flow does not effectively direct particles toward new modes; rather, it impedes their movement, often pulling them back into modes they have already explored. In contrast, the FHL algorithm demonstrates a remarkable ability to escape from metastable states and navigate these barriers efficiently. The corresponding results are illustrated in Figure 2.

5.1.2 EXPLOITATION

In our sampling process, we prioritize efficiently steering particles toward regions of high probability density, thereby avoiding unnecessary exploitation in low-probability areas. When sampling from a single image, our objective is to reach the global optimum, much like in typical optimization tasks.

For our experiment, we selected an image resembling the *GitHub* logo[1], converted it into a vector format, and used it as the mean of a multivariate Gaussian distribution. The covariance matrix for this distribution, denoted by Σ, is diagonal, with the variance for each dimension randomly drawn from a uniform distribution over the interval $(0.001, 1.0)$. Mathematically, the distribution is described by $e^{-U(x)} \propto \mathcal{N}(\mu, \Sigma)$, where $U(x)$ is the energy function that characterizes the system.

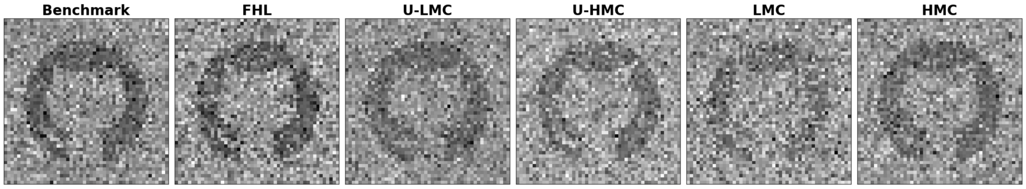


Figure 3: We obtain $N = 64$ samples from the Gaussian distribution $\mathcal{N}(\mu, \Sigma)$ (with $\mu \in \mathbb{R}^d$ representing the clean image) by running each method for $T = 256$ steps. From the resulting set X_T, we select and plot the sample with the lowest energy $U(x)$. The benchmark image is generated by directly sampling from $\mathcal{N}(\mu, \Sigma)$.

The results indicate that our FHL approach outperforms the baseline methods, particularly in scenarios where the energy landscape is ill-conditioned.

5.2 COMPOSITIONAL MODELS

A relationship between EBMs and score matching can be established by training EBMs through denoising score matching Song & Kingma (2021). The training objective is described below:

$$\mathcal{J}_\sigma(\theta) = \mathbb{E}_{x \sim \pi(x),\, \epsilon \sim \mathcal{N}(0,I)} \left[\left\| \frac{\epsilon}{\sigma} - \nabla_x U_\theta\big(x + \sigma\epsilon\big) \right\|_2^2 \right]. \tag{6}$$

Here, U_θ is typically parameterized by a neural network with parameters θ. Minimizing $\mathcal{J}_\sigma(\theta)$ enforces that $\nabla_x U_\theta(x) = -\nabla_x \log \pi_\sigma(x)$. So that $e^{-U_\theta(x)}$ is proportional to the smoothed target density

$$\pi_\sigma(x) = \int \pi(x')\, \mathcal{N}(x; x', \sigma^2 I)\, dx'. \tag{7}$$

As reported in Du et al. (2023), combining two diffusion models into a product model,

$$\pi^{\text{prod}}(x) \propto \pi^1(x)\pi^2(x), \tag{8}$$

can lead to issues if the reverse diffusion process simply sums the score estimates from the two independent models. In the following experiments (see Section 5.2.1 and Section 5.2.2), we demonstrate this issue using energy-based score-matching models.

5.2.1 SYNTHETIC DATASET

We begin by presenting an example of merging two distributions generated by DDPM (Ho et al., 2020). In this instance, the gradient does not consistently steer the particles into regions of high probability density.

The experimental results in Figure 4 demonstrate that FHL robustly converges to the desired composite distribution, with significantly fewer particles deviating from the high-density region compared to the baseline methods.

5.2.2 CLEVR DATASET

We employ the CLEVR dataset from Johnson et al. (2016) for our generation and sampling experiments, using the pre-trained energy model directly from Du et al. (2023). The dataset consists of three classes—*cube*, *sphere*, and *cylinder*—and we examine two scenarios: one in which samples are drawn from a single category, and another where samples are drawn from two categories.

[1] Downloaded from `https://github.com/logos`.

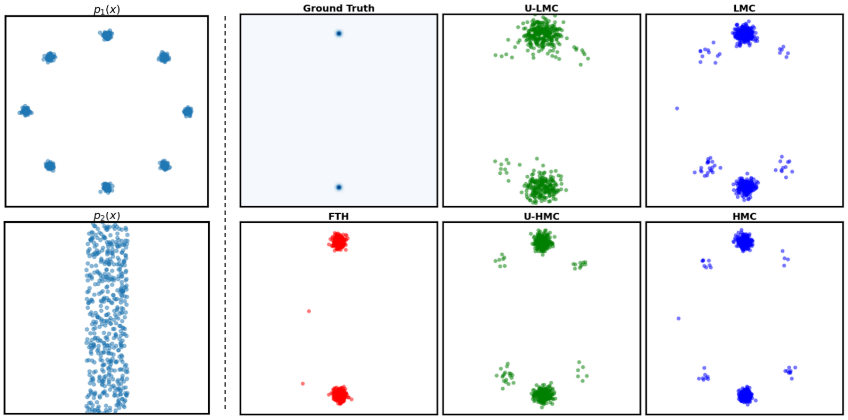
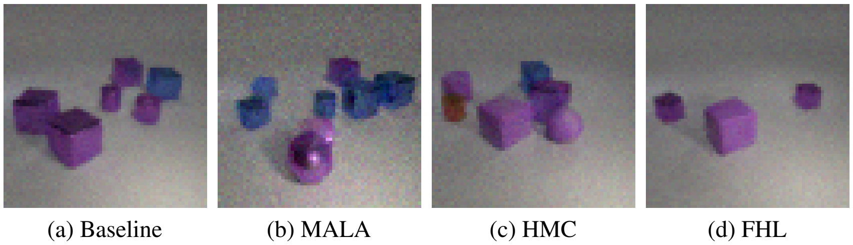


Figure 4: Compositional sampling from $\pi^{\text{prod}}(x) \propto \pi^1(x)\pi^2(x)$ is performed using DDPMs. In the leftmost column, samples from the original distributions π^1 and π^2 are displayed.

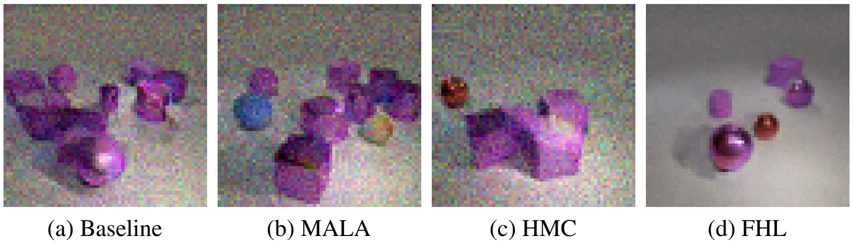
In the first experiment, no model composition is involved. As illustrated in Figure 5, FHL produces the target image without any extraneous shapes, whereas both MALA and HMC generate unwanted shapes. Furthermore, FHL exhibits reduced noise, indicating superior sampling quality.



(a) Baseline (b) MALA (c) HMC (d) FHL

Figure 5: Generation of *cube*.

In the second experiment, we combine two independent diffusion models, each trained separately to generate *sphere* and *cylinder*. As shown in Figure 6, it is clear that FHL excels at producing high-quality images with almost no overlapping between objects, accurately rendering the intended shapes in a pristine manner. In contrast, the other methods generate the undesired shape.



(a) Baseline (b) MALA (c) HMC (d) FHL

Figure 6: Generation of *sphere* and *cylinder*.

The experimental results demonstrate that FHL produces high-quality images, echoing our findings in Section 5.2.1 but under a more challenging setting. These results underscore the algorithm's robustness in handling complex data distributions while maintaining sampling quality.

6 Conclusion

In this study, we recognize the significance of incorporating zeroth-order information into the sampling process, highlighting the common limitations faced by conventional sampling methods. These limitations include unstable sampling outcomes frequently associated with energy-based score-matching models, the potential metastability arising from the multi-modal nature of the energy function, and errors in gradient computation stemming from the complex structure of the compositional distribution. Building upon HMC, we incorporate energy modulation techniques to enhance the sampling process. Through this approach, our method is able to systematically reduce the potential energy, leading to substantial advantages in practical implementations of sampling.

REFERENCES

Christophe Andrieu, Éric Moulines, and Francis J Samson. Particle markov chain monte carlo for efficient numerical simulation. *Statistical Science*, 25(4):332–350, 2010.

Michael Betancourt. A Conceptual Introduction to Hamiltonian Monte Carlo, July 2018.

Charles K. Birdsall and A. Bruce Langdon. *Plasma physics via computer simulation*. Taylor and Francis, New York, 2005. ISBN 0750310251 9780750310253.

Changyou Chen, David Carlson, Zhe Gan, Chunyuan Li, and Lawrence Carin. Bridging the gap between stochastic gradient mcmc and stochastic optimization. In Arthur Gretton and Christian C. Robert (eds.), *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pp. 1051–1060, Cadiz, Spain, 09–11 May 2016. PMLR. URL `https://proceedings.mlr.press/v51/chen16c.html`.

Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In Eric P. Xing and Tony Jebara (eds.), *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 1683–1691, Bejing, China, 22–24 Jun 2014a. PMLR. URL `https://proceedings.mlr.press/v32/cheni14.html`.

Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic Gradient Hamiltonian Monte Carlo. In *Proceedings of the 31st International Conference on Machine Learning*, pp. 1683–1691. PMLR, June 2014b.

Wenlin Chen, Mingtian Zhang, Brooks Paige, José Miguel Hernández-Lobato, and David Barber. Diffusive gibbs sampling. In *International Conference on Machine Learning*. PMLR, 2024.

Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Yilun Du, Conor Durkan, Robin Strudel, Joshua B. Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and mcmc. JMLR.org, 2023.

Tomas Geffner and Justin Domke. MCMC Variational Inference via Uncorrected Hamiltonian Annealing. In *Advances in Neural Information Processing Systems*, volume 34, pp. 639–651. Curran Associates, Inc., 2021.

Mark Girolami, Ben Calderhead, and Siu A Chin. Riemann Manifold Langevin and Hamiltonian Monte Carlo.

Ulf Grenander and Michael I. Miller. Representations of Knowledge in Complex Systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 56(4):549–603, 1994. ISSN 00359246.

W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. doi: 10.1093/biomet/57.1.97. URL `http://biomet.oxfordjournals.org/cgi/content/abstract/57/1/97`.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020.

Matthew D. Hoffman and Andrew Gelman. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo, November 2011.

Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005. URL `http://jmlr.org/papers/v6/hyvarinen05a.html`.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1988–1997, 2016. URL `https://api.semanticscholar.org/CorpusID:15458100`.

Yann LeCun, Sumit Chopra, Raia Hadsell, Marc Aurelio Ranzato, and Fu Jie Huang. A tutorial on energy-based learning. 2006.

Qiang Liu, Jason Lee, and Michael Jordan. A kernelized stein discrepancy for goodness-of-fit tests. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 276–284, New York, New York, USA, 20–22 Jun 2016. PMLR. URL `https://proceedings.mlr.press/v48/liub16.html`.

Yi-An Ma, Tianqi Chen, and Emily Fox. A Complete Recipe for Stochastic Gradient MCMC. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

Radford M. Neal. *MCMC Using Hamiltonian Dynamics*. May 2011. doi: 10.1201/b10905.

G. Parisi. Correlation functions and computer simulations. *Nuclear Physics B*, 180(3):378–384, 1981. ISSN 0550-3213. doi: 10.1016/0550-3213(81)90056-0.

Gareth O. Roberts and Richard L. Tweedie. Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341 – 363, 1996.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Jascha Sohl-Dickstein, Mayur Mudigonda, and Michael DeWeese. Hamiltonian monte carlo without detailed balance. In Eric P. Xing and Tony Jebara (eds.), *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 719–726, Bejing, China, 22–24 Jun 2014. PMLR. URL `https://proceedings.mlr.press/v32/sohl-dickstein14.html`.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2256–2265, Lille, France, 07–09 Jul 2015. PMLR. URL `https://proceedings.mlr.press/v37/sohl-dickstein15.html`.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Yang Song and Diederik P. Kingma. How to Train Your Energy-Based Models, February 2021.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=PxTIG12RRHS`.

Robert Swendsen and Jian-Sheng Wang. Replica monte carlo simulation of spin-glasses. *Physical review letters*, 57:2607–2609, 12 1986. doi: 10.1103/PhysRevLett.57.2607.

Luke Tierney. A note on metropolis-hastings kernels for general state spaces. *Annals of Applied Probability*, 8(1):1–9, 1998. doi: 10.1214/aoap/1027961031.

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

Follow Hamiltonian Leader: An Efficient Energy-Guided Sampling Method (Supplementary Material)

A Experiment Setup

In Section 5.2, we mainly adapted the codes and models from `https://github.com/yilundu/reduce_reuse_recycle`.

- In Section 5.2.1, we train a four-layer ResNet to serve as our energy-based score-matching model and use step sizes $\eta = \{0.002, 0.0002, 0.005, 0.0005\}$ for all methods..
- In Section 5.2.2, we employ a U-net architecture Ronneberger et al. (2015) as the energy-based score-matching model and conduct sampling with step sizes $\eta = \{0.01, 0.035, 0.05, 0.1, 0.2\}$ across all methods.

Additionally, we experiment with various configurations by varying the number of particles per group $n = \{2, 4, 8, 16\}$, testing different pulling strengths $\lambda = \{0.1, 1.0, 10.0\}$ and $\gamma = \{0.1, 0.2, 0.5, 0.9\}$, and trying $\sigma_l = \{0.1, 0.2, 0.5, 1.0\}$ along with $L = \{4, 8, 16\}$ for HMC-type sampling methods.

B Proof of Theorem 1

For clarity in our analysis, we first rewrite Algorithm 1 and Algorithm 2 in the following forms in Algorithm 3 and Algorithm 4. The Elastic function used in Algorithm 4 corresponds to a general case where the choice of λ in 3 can be made adaptive to the values of joint states.

Algorithm 3 FHL method per iteration t

```
Input: A joint state $s_t = (\{x_t^i\}_{i=1}^n, \{p_t^i\}_{i=1}^n) \in S$ at iteration $t$, learning rate $\eta > 0$, pulling strength $\lambda \geq 0$, number of Leapfrog steps $L$.
Step 1 (momentum resampling): $s'_t = R(s_t)$, where $R$ denotes the momentum resampling operation, i.e. $s'_t = R(s_t) = \left(\{x_t^i\}_{i=1}^n, \{p_{new}^i\}_{i=1}^n\right)$ where $\{p_{new}^i\}_{i=1}^n$ are i.i.d. samples drawn from the isotropic Gaussian distribution.
Step 2 (elastic eLeapfrog): From $s'_t$, eLeapfrog proposes a new state:
    $$s_+ = \left(\{x_+^i\}_{i=1}^n, \{p_+^i\}_{i=1}^n\right).$$
In other words, this step is a transform $\mathbf{L}$ such that $s_+ = \mathbf{L}(s'_t)$.
Step 3 (momentum flip, only in theory):
    $$s_{prop} = \mathbf{F}(s_+) = \left(\{x_+^i\}_{i=1}^n, \{-p_+^i\}_{i=1}^n\right).$$
Step 4: accept the proposed state $s_{prop}$ with probability
    $$\alpha = \min\left\{1, \frac{\pi(s_{prop})}{\pi(s)}\right\}.$$
If accepted, $s_{t+1} = s_{prop}$; otherwise, $s_{t+1} = s'_t$.
Step 5: leader pulling mechanism in #2 of Algorithm 2
output: $s_{t+1}$
```

Algorithm 4 eLeapfrog method per iteration k

```
**Input:** A joint state $s_k = (\{x_k^i\}_{i=1}^n, \{p_k^i\}_{i=1}^n) \in S$ at iteration $k$, learning rate $\eta > 0$, pulling strength $\lambda \geq 0$.
Step A: the (unique) leader $x_k^l$ is determined by $\{x_k^i\}_{i=1}^n$ , i.e.
    $x_k^l = \text{Leader}(\{x_k^i\}_{i=1}^n)$
From this, the elastic strength $\rho_k^i$ is computed collectively for each particle, i.e.
    $\rho_k^i = \text{Elastic}(\{x_k^i\}_{i=1}^n, x_k^l), \quad \forall i \leq n.$
Step B: update the momentum of each particle,
    $p_{k+1/2}^i = p_k^i - \frac{\eta}{2}(\nabla_x U(x_k^i) + \rho_k^i(x_k^i - x_k^l)), \quad \forall i \leq n.$
Step C: update the position of each particle,
    $x_{k+1}^i = x_k^i + \eta p_{k+1/2}^i, \quad \forall i \leq n.$
Step D: select the (unique) leader among $\{x_{k+1}^i\}_i$,
    $x_{k+1}^l = \text{Leader}(\{x_{k+1}^i\}_{i=1}^n)$
and then compute the elastic strength collectively,
    $\rho_{k+1}^i = \text{Elastic}(\{x_{k+1}^l\}_i, x_{k+1}^l), \quad \forall i \leq n$
Step E: update the momentum of each particle,
    $p_{k+1}^i = p_{k+1/2}^i - \frac{\eta}{2}(\nabla_x U(x_{k+1}^i) + \rho_{k+1}^i(x_{k+1}^i - x_{k+1}^l)), \quad \forall i \leq n.$
**output**: $s_{k+1} = (\{x_{k+1}^i\}_{i=1}^n, \{p_{k+1}^i\}_{i=1}^n)$
```

Let p_1 be the transition kernel of Step 1 in Algorithm 3, and $p_2(ds'|s)$ be the transition kernel of Steps 2-4 in Algorithm 3. Let p_3 be the transition kernel of Step 5 in Algorithm 3.

By the definition of p, we have

$$p(ds'|s) = \int p_3(ds'|s'')p_{12}(ds''|s), \quad p_{12}(ds'|s) = \int p_2(ds'|s'')p_1(ds''|s)$$

Note that the above integrals integrate the intermediate state s''. If s_t be the joint state at iteration t, then from Algorithm 3,

$$s_{t+1}|s_t' \sim p_2(\cdot|s_t')$$

According to Theorem 2 in Tierney (1998), p_2 corresponds to a standard Metropolis-Hastings algorithm with a deterministic proposal Q and an accept/reject function α. The proposal $Q(ds'|s)$ represents Steps 2-3, while the function $\alpha(s, s')$ equals to the accept/reject probability specified in Step 4.

Reversibility of eLeapfrog Note that the $\mathbf{L}$ in Step 2 of Algorithm 3 is composed of L repeated steps of a basic operation G in eLeapfrog, i.e. $\mathbf{L} = G^{\circ L}$. We first verify that

$$(\mathbf{F}G)^{-1} = \mathbf{F}G. \tag{9}$$

We assume that the operator G modifies the state $s_k = \left(\{x_k^i\}_{i=1}^n, \{p_k^i\}_{i=1}^n\right)$ in an internal loop of eLeapfrog summarized in Algorithm 4, which returns $s_{k+1} = G(s_k)$. The operator $\mathbf{F}$ applied to $G(s_k)$ will then flip the sign of the momentum p_{k+1}^i of each particle in s_{k+1}. As $s_k \in S$ is chosen arbitrarily, (9) is equivalent to

$$G\mathbf{F}(s_{k+1}) = \mathbf{F}(s_k), \tag{10}$$

because $(\mathbf{F}G)^{-1} = \mathbf{F}G$ is equivalent to $G\mathbf{F}G = \mathbf{F}$ due to $\mathbf{F}^{-1} = \mathbf{F}$.

To verify (10), we compute $G\mathbf{F}(s_{k+1})$ based on the steps A-E in Algorithm 4. The state $\mathbf{F}(s_{k+1})$ is $\left(\{x_{k+1}^i\}_{i=1}^n, \{-p_{k+1}^i\}_{i=1}^n\right)$. To apply G, we denote x_{k+1}^i by $\tilde{x}_k^i$ and $-p_{k+1}^i$ by $\tilde{p}_k^i$. We can now re-apply the steps A-E to the state of $\tilde{x}_k^i$ and $\tilde{p}_k^i$:

Step A. A unique and deterministic leader $\tilde{x}^l_k$ is determined by $\{\tilde{x}^i_k\}^n_{i=1}$, i.e.

$$\tilde{x}^l_k = \text{Leader}(\{\tilde{x}^i_k\}^n_{i=1})$$

Note that the leader being unique ensures that $\tilde{x}^l_k = x^l_{k+1}$. From this, the elastic strength $\tilde{\rho}^i_k$ is computed, i.e.

$$\tilde{\rho}^i_k = \text{Elastic}(\{\tilde{x}^i_k\}^n_{i=1}, \tilde{x}^l_k), \quad \forall i \leq n.$$

It is clear that $\tilde{\rho}^i_k = \rho^i_{k+1}$ for each $i \leq n$.

Step B. Update the momentum of each particle,

$$\tilde{p}^i_{k+1/2} = \tilde{p}^i_k - \frac{\eta}{2}(\nabla_x U(\tilde{x}^i_k) + \tilde{\rho}^i_k(\tilde{x}^i_k - \tilde{x}^l_k)), \quad \forall i \leq n.$$

As $\tilde{x}^i_k = x^i_{k+1}$, $\tilde{p}^i_k$, $\tilde{p}^i_k = -p^i_{k+1}$ and $\tilde{\rho}^i_k = \rho^i_{k+1}$, we have

$$\tilde{p}^i_{k+1/2} = -p^i_{k+1} - \frac{\eta}{2}(\nabla_x U(x^i_{k+1}) + \rho^i_{k+1}(x^i_{k+1} - \tilde{x}^l_{k+1})) = -p^i_{k+1/2}, \quad \forall i \leq n.$$

Step C. Update the position of each particle,

$$\tilde{x}^i_{k+1} = \tilde{x}^i_k + \eta\tilde{p}^i_{k+1/2} = x^i_{k+1} - \eta p^i_{k+1/2} = x^i_k, \quad \forall i \leq n.$$

Step D. Select the leader among $\tilde{x}^i_{k+1}$,

$$\tilde{x}^l_{k+1} = \text{Leader}(\{\tilde{x}^i_{k+1}\}^n_{i=1}) = x^l_k,$$

and then compute the elastic strength,

$$\tilde{\rho}^i_{k+1} = \text{Elastic}(\{\tilde{x}^l_{k+1}\}_i, \tilde{x}^l_{k+1}) = \rho^i_k, \quad \forall i \leq n$$

Step E. Update the momentum of each particle (verify as in *Step B*),

$$\tilde{p}^i_{k+1} = \tilde{p}^i_{k+1/2} - \frac{\eta}{2}(\nabla_x U(\tilde{x}^i_{k+1}) + \tilde{\rho}^i_{k+1}(\tilde{x}^i_{k+1} - \tilde{x}^l_{k+1})) = -p^i_k, \quad \forall i \leq n.$$

The above *A*-*E* steps show that effectively (10) holds.

Detailed balance of p_2 It reamains to check that p_2 has the detailed balance with respect to π^n, i.e.

$$p_2(ds'|s)\pi^n(ds) = p_2(ds|s')\pi^n(ds').$$

According to the condition (3) in Tierney (1998), it is sufficient to verify that:

$$\pi^n(ds)Q(ds'|s)\alpha(s, s') = \pi^n(ds')Q(ds|s')\alpha(s', s).$$

Since Q is a deterministic proposal, it is a Dirac measure

$$Q(ds'|s) = \delta_{\mathbf{FL}(s)}(ds').$$

As $\alpha(s, s') = \min\left\{1, \frac{\pi(s')}{\pi(s)}\right\}$, a key property is to verify that $(\mathbf{FL})^{-1} = \mathbf{FL}$ (so that each elastic Leapfrog operator G has a volume preservation property just like the original Leapfrog method). From (9), we have $G^{-1} = \mathbf{F}G\mathbf{F}$, and

$$(\mathbf{FL})^{-1} = (\mathbf{F} \circ G \circ G \circ \cdots \circ G)^{-1} = G^{-1} \circ \cdots G^{-1} \circ G^{-1} \circ \mathbf{F}^{-1}.$$

As $\mathbf{F}^{-1} = \mathbf{F}$ by definition, we have

$$\mathbf{F}G\mathbf{F} \circ \mathbf{F}G\mathbf{F} \circ \cdots \circ \mathbf{F}G\mathbf{F} \circ \mathbf{F}^{-1} = \mathbf{F} \circ G^{\circ L}.$$

This implies that we indeed have $(\mathbf{FL})^{-1} = \mathbf{FL}$. It follows from the special case 2.2 in Tierney (1998) that p_2 has the detailed balance.

Invariance of p Since the momentum resampling (Step 1) samples the marginal distribution of π^n, it preserves the invariance, i.e.,

$$\pi^n(ds') = \int p_1(ds'|s)\pi^n(s)\, ds.$$

As p_2 has the detailed balance, it also preserves the invariance, i.e.,

$$\pi^n(ds') = \int p_2(ds'|s)\pi^n(s)\, ds.$$

As a consequence, we conclude that p has the invariance property since p_3 is also a standard Metropolis-step, i.e.

$$\pi^n(ds') = \int p(ds'|s)\pi^n(s)\, ds.$$