

## **Sampling from Boltzmann densities with physics informed low-rank formats**

Paul Hagemann<sup>1</sup>, Janina Schütte<sup>2</sup>, David Sommer<sup>2</sup>,

Martin Eigel<sup>2</sup>, Gabriele Steidl<sup>1</sup>

submitted: December 17, 2024

<sup>1</sup> Technische Universität Berlin  
Straße des 17. Juni 135  
10623 Berlin  
Germany  
E-Mail: hagemann@math.tu-berlin.de  
steidl@math.tu-berlin.de

<sup>2</sup> Weierstrass Institute  
Mohrenstr. 39  
10117 Berlin  
Germany  
E-Mail: janina.schuette@wias-berlin.de  
david.sommer@wias-berlin.de  
martin.eigel@wias-berlin.de

No. 3153  
Berlin 2024



---

2020 *Mathematics Subject Classification.* 62F15, 65N75, 60H35, 65N22, 62H12, 65C05,

*Key words and phrases.* Approximate sampling, tensor trains, low rank formats, resampling, Langevin Monte Carlo.

P.H., M.E., G.S. & J.S. gratefully acknowledge the funding from DFG SPP 2298 "Theoretical Foundations of Deep Learning". M.E. & D.S. acknowledge support by the ANR-DFG project *COFNET: Compositional functions networks - adaptive learning for high-dimensional approximation and uncertainty quantification*. This study does not have any conflicts to disclose.

Edited by  
Weierstraß-Institut für Angewandte Analysis und Stochastik (WIAS)  
Leibniz-Institut im Forschungsverbund Berlin e. V.  
Mohrenstraße 39  
10117 Berlin  
Germany

Fax: +49 30 20372-303  
E-Mail: [preprint@wias-berlin.de](mailto:preprint@wias-berlin.de)  
World Wide Web: <http://www.wias-berlin.de/>

# Sampling from Boltzmann densities with physics informed low-rank formats

Paul Hagemann, Janina Schütte, David Sommer,  
Martin Eigel, Gabriele Steidl

## Abstract

Our method proposes the efficient generation of samples from an unnormalized Boltzmann density by solving the underlying continuity equation in the low-rank tensor train (TT) format. It is based on the annealing path commonly used in MCMC literature, which is given by the linear interpolation in the space of energies. Inspired by Sequential Monte Carlo, we alternate between deterministic time steps from the TT representation of the flow field and stochastic steps, which include Langevin and resampling steps. These adjust the relative weights of the different modes of the target distribution and anneal to the correct path distribution. We showcase the efficiency of our method on multiple numerical examples.

## 1 Introduction

Sampling from Boltzmann densities with unknown normalization constant is an important and timely research problem [1, 9, 25]. The main challenge is that no samples are given. It hence is difficult to explore the probability space where the mass is concentrated. In the context of MCMC, this leads to the so-called energy barriers [4], which means that it takes chains a long time to jump from one mode of the distribution to another with uninformed proposals. In the context of (conditional) normalizing flows, the training of sampling problems is usually carried out using the reverse Kullback-Leibler (KL) divergence [25], which leads to unwanted mode-collapse. This in principle can be fixed by using MCMC steps to either favor exploration via path learning ideas [3, 15, 37] or by considering a mode-covering objective [21].

Recently, the path-based learning objective literature has not only been explored via variational objectives, which essentially rely on reformulations of the KL divergence, but also by learning the distributions via the corresponding PDEs. This can for instance be the continuity equation [19] or the Hamilton-Jacobi-Bellman equation [5]. Conceptionally, it is often done in a way similar to physics-informed neural networks (PINN) [29], which is known to suffer from stability and optimization issues [7] as well as slow training.

In this work, we address these shortcomings. Concretely, we learn the velocity field of the continuity equation inspired by [19], which was refined in [34]. To overcome stability issues and speed up training, we explore the tensor train (TT) framework [18, 26, 27], which has shown great success in solving PDEs [13, 11, 30, 36], and which has recently been used for sampling with diffusion paths [33]. Here, we follow a classical approach often used in Sequential Monte Carlo methods [10, 12, 23] and reformulate the (discrete) continuity equation by expansion into a product basis. This reduces training to solving linear systems of equations. To facilitate expressivity, we enrich the flow with stochastic steps combining ideas from [3, 12, 37], in particular resampling and Langevin steps. In contrast to [3], we replace their normalizing flow layers with TT constructions with associated functional basis.

Our contributions can be summarized as follows:

- We introduce a functional tensor train (FTT) format for solving the continuity equation for the annealing path. In particular, we derive formulas for the action of the continuity equation on vector fields represented by FTTs with arbitrary basis functions.
- Using the action of the continuity equation on FTT, we propose a numerical optimization scheme involving empirical risk minimization with the alternating linear scheme (ALS) [17]. Our approximation is naturally rank-adaptive due to an exponential moving average of successive TT approximations.
- We apply an  $H^2$  (Sobolev) orthonormal Fourier basis opposed to the commonly used polynomials within the TT format [11, 14, 30].
- We add stochastic steps, namely Langevin postprocessing and resampling, to improve expressivity.
- We illustrate the efficiency of the format on two-dimensional Gaussian mixture problems and the “many well problem” introduced in [21].

## 2 Logarithmic Continuity Equation

We want to sample from Boltzmann densities depending on functions  $f : [0, 1] \times \mathbb{R}^d \rightarrow \mathbb{R}$  satisfying mild conditions such that for  $f_t(\cdot) := f(t, \cdot)$

$$p_t := \frac{e^{-f_t}}{Z_t}, \quad Z_t := \int_{\mathbb{R}^d} e^{-f_t} dx.$$

We follow the considerations in [6, 19] and refer to [6] for a mathematically rigorous treatment. Starting with an easy-to-sample latent distribution  $p_0$ , the goal is to sample from a target distribution  $p_1$  knowing only  $f_1$ . In this paper, we follow the annealing path

$$f_t = tf_1 + (1 - t)f_0, \quad t \in [0, 1],$$

which is commonly used in the MCMC literature [3, 12, 23, 37]. Now the task is to find a measurable function  $v : [0, 1] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$  such that  $(p_t, v_t)$  satisfies the continuity equation

$$\partial_t p_t + \nabla \cdot (p_t v_t) = 0.$$

For the Boltzmann density, the continuity equation becomes [19]

$$(\partial_t f_t + \langle \nabla f_t, v_t \rangle - \nabla \cdot v_t + C_t) p_t = 0 \quad \text{with} \quad C_t := \partial_t (\log Z_t).$$

With the linear interpolation (2) we obtain

$$f_1 - f_0 + \langle \nabla f_t, v_t \rangle - \nabla \cdot v_t + C_t = 0 \quad \text{a.e. } t \in [0, 1].$$

Indeed, it was shown under mild assumptions on  $p_0$  and  $p_1$  that appropriate Borel measurable vector fields  $v_t$  exist, see [6] and the references therein. Fortunately, absolutely continuous curves  $p_t$  with

corresponding vector fields  $v_t$  in (2) can be described by the solution  $\phi : [0, 1] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$  of an ODE

$$\partial_t \phi(t, x) = v_t(\phi(t, x)), \quad \phi(0, x) = x$$

and  $\phi(t, \cdot) \# p_0 = p_t$ , see [2, Theorem 8.1.8]. Therefore, once the velocity field satisfying (2) is known, one can sample from  $p_1$  by solving the above ODE with classical numerical solvers. In this paper, we propose to approximate  $v_t$  by a low-rank TT representation  $v_t^\theta$  that minimizes by (2) the least squares problem

$$\mathcal{J}_t(\theta) := |f_1 - f_0 + \langle \nabla f_t, v_t^\theta \rangle - \nabla \cdot v_t^\theta + C_t|^2.$$

### 3 Low-rank tensor train decomposition

We propose to approximate the vector field  $v_t$  with a tensor product ansatz. Choosing a basis  $\{\varphi_i : \mathbb{R} \rightarrow \mathbb{R} : i \in [n]\}$ ,  $[n] := \{1, \dots, n\}$ , of a space of univariate functions,  $v_t^\theta : \mathbb{R}^d \rightarrow \mathbb{R}^d$  is then represented in a product basis expansion. Collecting the basis coefficients in an array yields a high-dimensional tensor  $T \in \mathbb{R}^{d \times n \times \dots \times n}$  with  $d n^d$  entries. Hence, the storage complexity quickly becomes prohibitive in higher dimensions. To alleviate this, an approximation is directly learned in a lower dimensional tensor manifold. There are different low-rank formats used in the literature, such as the canonic-polyadic (CP), the hierarchical (HT) or TT format. For a comprehensive overview, we refer to [24]. The TT format is particularly useful for approximation tasks: contrary to the CP format, the set of tensors with so-called ‘‘TT rank’’ smaller than a fixed upper bound is closed. Moreover, there exists a gradient-free alternating optimization procedure known as *alternating linear scheme* (ALS), which is used in this work.

For a formal derivation of the TT format, we note that each entry of a tensor  $T \in \mathbb{R}^{d \times n \times \dots \times n}$  can be represented by a multiplication of matrices. For a multi-index  $(i_1, \dots, i_d) \in [n] \times \dots \times [n]$ , we write

$$T_{:,i_1,\dots,i_d} = K_{i_1}^{(1)} \cdot \dots \cdot K_{i_d}^{(d)} \in \mathbb{R}^d,$$

where  $K_{i_1}^{(1)} \in \mathbb{R}^{d \times r_1}$ ,  $K_{i_d}^{(d)} \in \mathbb{R}^{r_{d-1} \times 1}$  and  $K_{i_k}^{(k)} \in \mathbb{R}^{r_{k-1} \times r_k}$  for  $k = 2, \dots, d-1$  are matrices. The smallest values  $r_1, \dots, r_{d-1} \in \mathbb{N}$  such that a representation (3) exists are called the *TT-ranks*. The three dimensional tensors  $K^{(k)} \in \mathbb{R}^{r_{k-1} \times n \times r_k}$  containing the matrices  $K_{i_k}^{(k)}$  for  $k \in [d]$  and  $i_k \in [n]$  are called the *cores* (or *component tensors*) of the TT. Using a graphical notation, we can write the tensor as a contraction of the TT cores along the rank dimensions

$$\begin{array}{c} n \\ | \\ \text{---} T \text{---} \\ | \\ d \end{array} \longrightarrow \begin{array}{c} d \\ | \\ \text{---} K^{(1)} \text{---} \\ | \\ n \end{array} \begin{array}{c} r_1 \\ | \\ \text{---} K^{(2)} \text{---} \\ | \\ n \end{array} \dots \begin{array}{c} r_{d-1} \\ | \\ \text{---} K^{(d)} \text{---} \\ | \\ n \end{array}.$$

The basis expansion with coefficient tensor  $T$  can be rewritten based on the TT representation for any  $x = (x_1, \dots, x_d)^\top \in \mathbb{R}^d$  as follows:

$$v(x) = \sum_{i_1=1}^{n_1} \dots \sum_{i_d=1}^{n_d} T_{:,i_1,\dots,i_d} \prod_{k=1}^d \varphi_{i_k}(x_k) \longrightarrow v(x) = \sum_{i_1=1}^{n_1} \dots \sum_{i_d=1}^{n_d} \prod_{k=1}^d K_{i_k}^{(k)} \varphi_{i_k}(x_k).$$

Imposing a low-rank structure on the coefficient tensor  $T$  results in a manifold of functions with low-rank tensors of rank at most  $\mathbf{r} = (r_1, \dots, r_{N-1})$ . In practice this can be understood as neglecting spurious features by an implicit feature selection. Analytical bounds, which quantify this behavior rigorously are given in [26]. Explicit constructions of some notable functions in TT format can be found in [27]. The combination of a TT with an associated set of basis functions defines an FTT.

### 3.1 Vector field learning with the alternating linear scheme

We approximate the vector field  $v_t$  at fixed times  $t$  as an FTT  $v_t^\theta$  of order  $d$  and with  $d$ -dimensional output. The parameter  $\theta$  is short-hand notation for the collection of all TT components  $K_t^{(1)}, \dots, K_t^{(d)}$  defining  $v_t^\theta$ . The FTT is the contraction of the TT components with the basis functions  $\Phi(x_k) := (\varphi_1(x_k), \dots, \varphi_n(x_k))^T$  for each  $k \in [d]$ , yielding a  $d$ -dimensional output,

$$v_t^\theta(x) = \overset{d}{\underbrace{K_t^{(1)}}_{n \mid}} \overset{r_1}{\text{---}} \overset{r_1}{\underbrace{K_t^{(2)}}_{n \mid}} \overset{r_2}{\text{---}} \dots \overset{r_{d-1}}{\text{---}} \overset{r_{d-1}}{\underbrace{K_t^{(d)}}_{n \mid}} \cdot$$

$\Phi(x_1)$

$\Phi(x_2)$

$\Phi(x_d)$

For the training process, the empirical  $L^2$  error in (2) on points  $\{x^{(\ell)}\}_{\ell=1}^N$  with

$$J_{t,N}(\theta) = \sum_{\ell=1}^N \left| \partial_t f_t(x^{(\ell)}) + \langle \nabla f_t(x^{(\ell)}), v_t^\theta(x^{(\ell)}) \rangle - \nabla \cdot v_t^\theta(x^{(\ell)}) + C_t \right|^2$$

is minimized. In the alternating linear scheme [17], the mappings  $K_t^{(i)} \mapsto J_{t,N}(\theta)$  are iteratively minimized over  $K_t^{(i)}$ , while keeping the other components fixed. Once  $K_t^{(i)}$  is optimized, the next component  $K_t^{(i+1)}$  is optimized and so on, sweeping forward and backward through the chain of TT components until a (local) minimum is reached.

By (3.1), there exist linear operators  $\mathcal{L}_i: \mathbb{R}^{r_{i-1} \times n \times r_i} \rightarrow \mathbb{R}^N$  such that the minimization over one core can be written as

$$\min_{K_t^{(i)}} J_{t,N}(v_t^\theta) = \min_{K_t^{(i)}} \left\| \mathcal{L}_i(K_t^{(i)}) - y \right\|^2,$$

where  $y \in \mathbb{R}^N$  with  $y_\ell = -\partial_t f_t(x^{(\ell)}) - C_t$  and the entries of  $\mathcal{L}_i(K_t^{(i)})$  are defined for  $\ell = 1, \dots, N$  in graphical notation by

$$\mathcal{L}_i(K_t^{(i)})_\ell = \sum_{k=1}^d \left[ \overset{r_1}{\underbrace{K_t^{(1)}[k]}_{n_1 \mid}} \overset{r_1}{\text{---}} \dots \overset{r_{k-1}}{\text{---}} \overset{r_{k-1}}{\underbrace{K_t^{(k)}}_{n \mid}} \overset{r_k}{\text{---}} \dots \overset{r_{i-1}}{\text{---}} \overset{r_{i-1}}{\underbrace{K_t^{(i)}}_{n \mid}} \overset{r_i}{\text{---}} \dots \overset{r_{d-1}}{\text{---}} \overset{r_{d-1}}{\underbrace{K_t^{(d)}}_{n \mid}} \right] \cdot$$

$\Phi(x_1^{(\ell)})$

$(\partial_k f_t(x^{(\ell)}) - \partial_k) \Phi(x_k^{(\ell)})$

$\Phi(x_i^{(\ell)})$

$\Phi(x_d^{(\ell)})$

Here,  $K_t^{(1)}[k]$  denotes the  $\mathbb{R}^{1 \times r_1}$ -vector defined by  $(K_t^{(1)}[k])_{i,j} := K_t^{(1)}_{k,i,j}$ . Due to the linearity of  $\mathcal{L}_i$ , (3.1) is a simple quadratic program, which can be solved by standard techniques. The ALS solves this quadratic program iteratively for all components until convergence. The convergence of ALS to local minima is investigated in [31].

## 4 Methodology

The vector field  $v_t = v_t^\theta$  is learned on a fixed time grid  $t_0 = 0, \dots, t_m = m \cdot \frac{1}{M}, \dots, t_M = 1$  by the procedure described in Section 3.1. In the following, we discuss the critical question of how to choose the samples defining the empirical loss (3.1) at any given time.

## 4.1 Stochastic steps

There are two main obstacles when solving the PDE (3.1). First, it is unclear how to choose the samples on which to solve the PDE. We could use uniform samples in space. However, this scales unfavorably. Alternatively, we can follow [19], which always draws from the current trained flow. This is similar to a *mode-seeking* reverse KL in the context of normalizing flows [28] and our method of choice. Pushing samples along the trajectories of the learned vector field is later denoted by `ODEsolve`. Second, the path of  $f_t$  is well-known to exhibit large jumps in the velocity field close to the final time, where relative mode weights get reassigned. This is called “teleportation issue” in [6], where it is investigated analytically. See also [19] for visualizations of this issue.

Both obstacles can be mitigated by adding stochastic steps. This has been crucial in many sampling related works, such as [3, 16, 37]. After the TT steps, two kinds of stochastic steps are added as outlined in Algorithm 1:

- *Langevin steps*: Given an energy  $f$  of some density  $p$ , we can sample asymptotically from  $p$  by following the dynamics

$$X_{\ell+1} = X_{\ell} - h \nabla f(X_{\ell}) + \sqrt{2h} \mathcal{N}(0, I),$$

where  $(X_{\ell})_{\ell}$  denotes a Markov chain that converges asymptotically to samples from  $p \sim e^{-f}$  [8, 22, 38]. Required for this are growth and regularity conditions on  $f$  such as strong convexity and Lipschitz gradient of the energy. Only a few steps of this update rule are applied and denoted by `LangevinStep` in Algorithm 1.

- *Resampling steps*: Since Langevin sampling struggles with mode mixing [37], we also perform so-called “resampling steps” as commonly used in Sequential Monte Carlo methods [3, 10, 12]. Here, we are given some points  $\{x^{(\ell)}\}_{\ell=1}^N$  drawn from some density  $q$ , which can be evaluated up to the normalizing constant. Assume we target density  $p$ . Then, for each of the points  $x^{(\ell)}$  the importance weights [23]  $w^{(\ell)} = \frac{p(x^{(\ell)})}{q(x^{(\ell)})}$  are calculated. Then, samples are drawn from the categorical distribution proportional to  $\sum_{\ell} w^{(\ell)} x^{(\ell)}$ . This however yields the “same” sample multiple times, which is why we also add Langevin steps after resampling steps. This procedure is denoted by `ResamplingStep`.

While the resampling steps help to overcome teleportation issues, the Langevin dynamics deter a collapse of the resampling steps to point masses. For the resampling step, we still have to discuss how  $q$  should be chosen. Advancing step  $k \rightarrow k+1$  with a known transport  $T_k = \phi(t_{k+1}, \cdot) \circ \phi(t_k, \cdot)^{-1}$  with  $(T_k)_{\#} p_k = q \approx p_{k+1}$  as in (2), the density  $q$  can be written as  $q(x) = p_k(T_k^{-1}(x)) |\det \nabla T_k^{-1}(x)|$  by leveraging the change of variables formula. The map  $T_k$  given a velocity field  $v$  can be defined by  $T_k(x) = x + \int_{t_k}^{t_{k+1}} v_s(x_s) ds$ , which we approximate using our discrete formulation (and similarly for  $T_k^{-1}$ ). In practice we find it beneficial to use Langevin steps before and after resampling steps. However, assessing the effect of the Langevin layers on the density  $q$  is left for future work and we only use the flow layers.

## 4.2 Adaptive TT ranks

The training on one set of prior samples, which is transported by the vector field to serve as training samples for the TT in the next time step, is repeated to train the TTs on more samples. In each iteration,

the distribution of the samples in later time steps should be closer to the interpolated distribution, since the TTs approximate the solution of the PDE increasingly better in each iteration. They therefore serve as better training samples. When training on a new set of samples, the ALS updates each core without consideration of the information from the last iteration. To not deviate too far from the previously learned TT in the last iteration, the old and newly learned TTs are added in a weighted manner similar to an *exponential moving average (EMA)*. The TT representing the addition can exhibit doubled ranks, which can be recompressed by applying a singular value decomposition on each core and truncating small singular values (TT-SVD, [26]). This procedure is denoted by `UpdateTT` in Algorithm 1 and bears similarities with the CRAFT framework [20]. By this, the ranks of the TT do not have to be set a priori but are adjusted dynamically during training. The resulting algorithm is described in Algorithm 1.

---

**Algorithm 1:** Training the vector field
 

---

```

Input: latent density  $p_0$ , target energy  $f_1$ 
while not converged do
  sample prior  $x \leftarrow \{x_i\}_{i \in [N]} \sim p_0$  ▷ initialize training data set
  for  $k = 1, \dots, M$  do
    calculate  $\partial_t f_{t_k}(x), \nabla f_t(x)$ 
    train  $v_{t_k}^{\text{new}}$  with ALS to minimize (3.1) ▷ train vector field
     $v_{t_k} \leftarrow \text{UpdateTT}(v_{t_k}, v_{t_k}^{\text{new}})$  ▷ update and adjust TT ranks
     $x \leftarrow \text{ODEsolve}(x, t_k, t_{k+1}, v_{t_k})$  ▷ update samples for next time step
     $x \leftarrow \text{ResamplingStep}(x)$ 
     $x \leftarrow \text{LangevinStep}(x)$ 
  end
end
return:  $v_{t_1}, \dots, v_{t_M}$ 

```

---

## 5 Numerical experiments

The proposed architecture is tested with different target distributions. For comparison with other works, the energy distance as in [32, 35] is used.

**Target Distributions.** As targets, we choose the following three distributions:

- 1 *Gaussian mixture with two modes:* The two-dimensional Gaussian mixture is considered with two modes. Both have same weights and variance 0.01. One Gaussian is with mean  $(2, 2)$  and other with mean  $(-2, -2)$ .
- 2 *Gaussian mixture with 40 modes:* The two-dimensional Gaussian mixture as in [6, 21] is considered, where 40 different normal distributions with means uniformly distributed in  $[-40, 40]^2$  are mixed with equal weights.
- 3 *Many well problem (MWP):* The target density in the  $d$ -dimensional MWP as in [6, 21] consists of the multiplication of  $m$  copies of the two-dimensional double well potential and  $d - 2m$  Gaussians given by

$$f_1^{m,d} = \sum_{i=0}^m (x_{2i}^4 - 6x_{2i}^2 - \frac{1}{2}x_{2i} + \frac{1}{2}x_{2i+1}^2) + \sum_{i=2m+1}^d \frac{1}{2}x_i^2.$$



Table 1: Average energy distance  $\pm$  standard deviation over 100 independent draws of 50.000 samples for the methods flow, flow+, flow+stochastic and stochastic. Best method in bold.

	flow	flow+	flow+stochastic	stochastic
GM 2 modes	$3.7\text{e-}3 \pm 2\text{e-}4$	<b><math>1.2\text{e-}3 \pm 5\text{e-}4</math></b>	<b><math>1.1\text{e-}3 \pm 1\text{e-}3</math></b>	$2.7\text{e-}3 \pm 3\text{e-}3$
GM 40 modes	$4.5\text{e-}1 \pm 3\text{e-}2$	<b><math>7.6\text{e-}2 \pm 2\text{e-}2</math></b>	<b><math>7.6\text{e-}2 \pm 4\text{e-}2</math></b>	$1.4\text{e-}1 \pm 9\text{e-}2$
Many well $f_1^{4,8}$	$1.6\text{e-}1 \pm 1\text{e-}4$	$1.2\text{e-}2 \pm 2\text{e-}4$	<b><math>1.9\text{e-}3 \pm 2\text{e-}4</math></b>	$8.1\text{e-}2 \pm 9\text{e-}3$
MWP $f_1^{4,16}$	$2.1\text{e-}1 \pm 2\text{e-}3$	$1.3\text{e-}1 \pm 6\text{e-}3$	<b><math>6.6\text{e-}3 \pm 2\text{e-}3</math></b>	$7.4\text{e-}2 \pm 1\text{e-}2$

**Methods.** To assess the importance of the different algorithmic steps, we test the following methods:

- **flow:** a pure TT normalizing flow is trained without stochastic steps.
- **flow+:** after the full TT flow, resampling and Langevin steps are carried out.
- **flow+stochastic:** a flow is trained with stochastic steps in the training process and the evaluation process after every time step as in Algorithm 1.
- **stochastic:** the results are compared with samples generated by the same amount of stochastic steps without steps of a trained flow, where we take the ratio for the resampling step to be  $\frac{p_{k+1}}{p_k}$ .

In all examples, the ground truth is known. The models are tuned by comparison with a validation set of samples. The tuning procedures without ground truth samples is left for future work.

In all experiments, the basis is chosen to be an  $H^2$  orthonormalized Fourier basis and the number of time steps is  $M \leq 35$ . The number of Fourier basis coefficients is  $n \leq 10$  in all experiments except for Problem 2, in which we use  $n = 41$ . In Problems 1 and 3, the basis is orthonormalized on  $[-5, 5]^d$  and the latent distribution is chosen to be a normal distribution. In Problem 2, the parameters are  $[-50, 50]^2$  and we start in a zero mean Gaussian with standard deviation 500.

Table 1 contains the results for the four methods and four different target distributions. It can be seen that the pure flow already achieves a good fit. Especially in higher dimensions, the stochastic steps are needed to achieve very good performance. Usually either *flow+* or *flow+stochastic* take the lead, where *flow+stochastic* seems to perform especially well in the high-dimensional MWP examples. We also see that the flow improves upon the stochastic steps since *flow+stochastic* usually outperforms *stochastic*. We show the marginal distributions of the 4 methods in Figure 5.2 for some marginals of the MWP. Here, continuous TT flow (*flow*) struggles to separate the mass perfectly, which is resolved by the stochastic steps. Figure 5.1 depicts the samples generated for the 40-mode GM problem.

## 6 Conclusions

In this work, the continuity equation of the annealing path is solved in a functional low-rank TT format to draw samples from a given target density. In contrast to other works, a Fourier basis orthogonalized with respect to the  $H^2$  scalar product is used for the FTT architecture. This choice works well to solve the PDE, while common basis functions such as polynomials perform poorly. This observation could be attributed to the lack of Lipschitz continuity of higher-order polynomials and will be examined in future work. The TT architecture is adapted during training similar to an EMA, leading to a more stable training procedure and the advantage that the architecture does not have to be set a priori. In

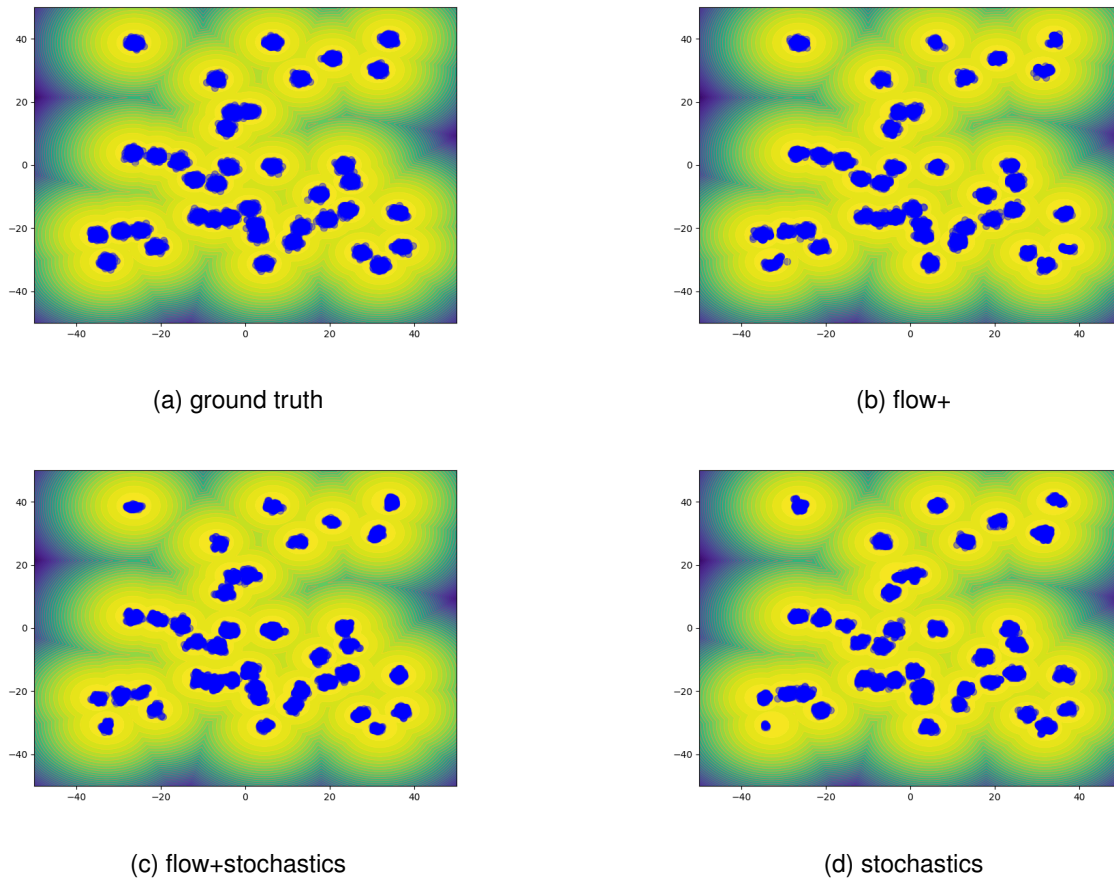


Figure 5.1: GM density sampled with the different methods. While the methods perform similarly on this problem, the TT flow in (b) and (c) helps to distribute the mass correctly compared to a purely stochastic approach (as can e.g. be seen for the mode on the lower left part of the domain).

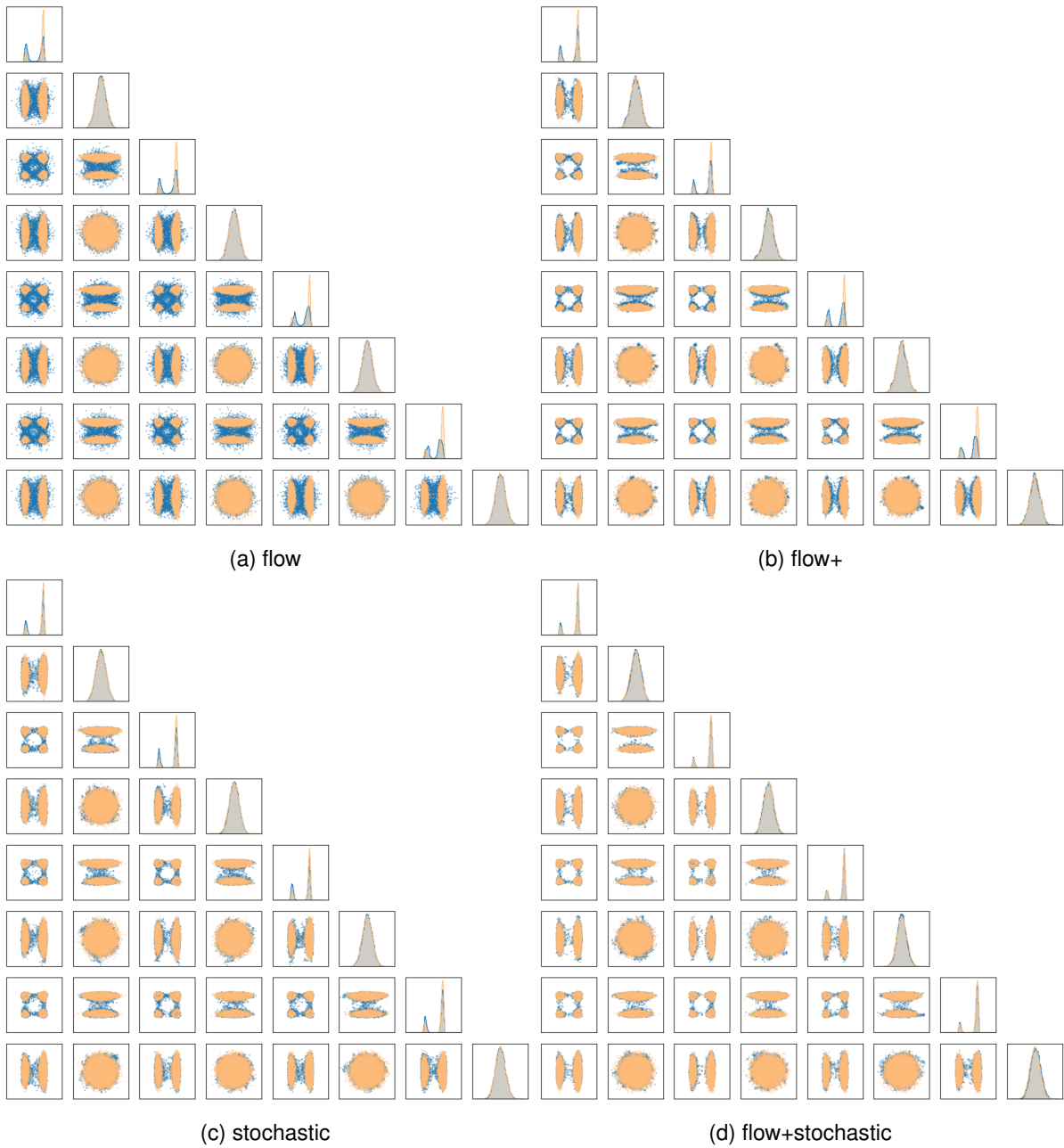


Figure 5.2: Histogram corner plots showing the 1-d marginals on the diagonal and the 2-d joint distributions on the off-diagonal for the MWP  $f_1^{4,16}$  for different methods (blue: competing method, orange ground truth).

future work, non-sample-based techniques could be developed to choose the best architecture during training, e.g., based on the residual of the PDE.

## References

- [1] M. Albergo, G. Kanwar, and P. Shanahan. Flow-based generative models for Markov chain Monte Carlo in lattice field theory. *Physical Review D*, 100(3), Aug. 2019.
- [2] L. Ambrosio, N. Gigli, and G. Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2008.
- [3] M. Arbel, A. Matthews, and A. Doucet. Annealed flow transport Monte Carlo. In *International Conference on Machine Learning*, pages 318–330. PMLR, 2021.
- [4] A. Bandeira, A. Maillard, R. Nickl, and S. Wang. On free energy barriers in gaussian priors and failure of cold start MCMC for high-dimensional unimodal distributions. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 381:20220150, 03 2023.
- [5] J. Berner, L. Richter, and K. Ullrich. An optimal control perspective on diffusion-based generative modeling. *Transactions on Machine Learning Research*, 2024.
- [6] J. Chemseddine, C. Wald, R. Duong, and G. Steidl. Neural sampling from Boltzmann densities: Fisher-Rao curves in the Wasserstein geometry. *arXiv preprint arXiv:2410.03282*, 2024.
- [7] P.-Y. Chuang and L. A. Barba. Experience report of physics-informed neural networks in fluid simulations: pitfalls and frustration. *arXiv preprint arXiv:2205.14249*, 2022.
- [8] A. S. Dalalyan and A. G. Karagulyan. User-friendly guarantees for the Langevin Monte Carlo with inaccurate gradient. *ArXiv*, abs/1710.00095, 2017.
- [9] L. Del Debbio, J. Marsh Rossney, and M. Wilson. Efficient modeling of trivializing maps for lattice  $\phi^4$  theory using normalizing flows: A first look at scalability. *Physical Review D*, 104(9), Nov. 2021.
- [10] P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society Series B: Statist. Method.*, 68(3):411–436, 2006.
- [11] S. Dolgov, D. Kalise, and K. K. Kunisch. Tensor decomposition methods for high-dimensional Hamilton–Jacobi–Bellman equations. *SIAM Journal on Scientific Computing*, 43(3):A1625–A1650, 2021.
- [12] A. Doucet, S. Godsill, and C. Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10:197–208, 2000.
- [13] M. Eigel, J. Neumann, R. Schneider, and S. Wolf. Non-intrusive tensor reconstruction for high-dimensional random pdes. *Computational Methods in Applied Mathematics*, 19(1):39–53, 2019.
- [14] M. Eigel, R. Schneider, and D. Sommer. Dynamical low-rank approximations of solutions to the Hamilton–Jacobi–Bellman equation. *Numerical Linear Algebra with Applications*, 30(3):e2463, 2023.

- [15] P. Hagemann, J. Hertrich, and G. Steidl. Stochastic normalizing flows for inverse problems: A Markov chains viewpoint. *SIAM/ASA Journal on Uncertainty Quantification*, 10(3):1162–1190, 2022.
- [16] J. Hertrich and R. Gruhlke. Importance corrected neural JKO sampling. *arXiv preprint arXiv:2407.20444*, 2024.
- [17] S. Holtz, T. Rohwedder, and R. Schneider. The alternating linear scheme for tensor optimization in the tensor train format. *SIAM Journal on Scientific Computing*, 34(2):A683–A713, 2012.
- [18] S. Holtz, T. Rohwedder, and R. Schneider. On manifolds of tensors of fixed TT-rank. *Numerische Mathematik*, 120(4):701–731, 2012.
- [19] B. Máté and F. Fleuret. Learning interpolations between boltzmann densities. *Transactions on Machine Learning Research*, 2023.
- [20] A. G. D. G. Matthews, M. Arbel, D. J. Rezende, and A. Doucet. Continual repeated annealed flow transport Monte Carlo. In *Proceedings of the 39th Int. Conference on Machine Learning*, Proceedings of Machine Learning Research, Jul 2022.
- [21] L. I. Midgley, V. Stimper, G. N. C. Simm, B. Schölkopf, and J. M. Hernández-Lobato. Flow annealed importance sampling bootstrap. *arXiv preprint arXiv:2208.01893*, 2023.
- [22] C. Miranda, J. Schütte, D. Sommer, and M. Eigel. Approximating Langevin Monte Carlo with ResNet-like neural network architectures. *arXiv preprint arXiv:2311.03242*, 2024.
- [23] R. M. Neal. Annealed importance sampling. *Statist. and Comp.*, 11:125–139, 2001.
- [24] A. Nouy. *Chapter 4: Low-Rank Methods for High-Dimensional Approximation and Model Order Reduction*, page 171–226. Society for Industrial and Applied Mathematics, July 2017.
- [25] F. Noé, S. Olsson, J. Köhler, and H. Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457), 2019.
- [26] I. V. Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.
- [27] I. V. Oseledets. Constructive representation of functions in low-rank tensor formats. *Constructive Approximation*, 37:1–18, 2013.
- [28] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.
- [29] M. Raissi, P. Perdikaris, and G. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Comp. Phys.*, 378:686–707, 2019.
- [30] L. Richter, L. Sallandt, and N. Nüsken. Solving high-dimensional parabolic PDEs using the tensor train format. In *International Conference on Machine Learning*, pages 8998–9009. PMLR, 2021.
- [31] T. Rohwedder and A. Uschmajew. On local convergence of alternating schemes for optimization of convex problems in the tensor train format. *SIAM Journal on Numerical Analysis*, 51(2):1134–1162, 2013.

- [32] D. Sejdinovic, B. Sriperumbudur, A. Gretton, and K. Fukumizu. Equivalence of distance-based and RKHS-based statistics in hypothesis testing. *The Annals of Statistics*, 41(5):2263 – 2291, 2013.
- [33] D. Sommer, R. Gruhlke, M. Kirstein, M. Eigel, and C. Schillings. Generative modelling with tensor train approximations of Hamilton–Jacobi–Bellman equations. *arXiv preprint arXiv:2402.15285*, 2024.
- [34] J. Sun, J. Berner, L. Richter, M. Zeinhofer, J. Müller, K. Azizzadenesheli, and A. Anandkumar. Dynamical measure transport and neural PDE solvers for sampling. *arXiv preprint arXiv:2407.07873*, 2024.
- [35] G. J. Székely and M. L. Rizzo. Energy statistics: A class of statistics based on distances. *Journal of Statistical Planning and Inference*, 143(8):1249–1272, 2013.
- [36] S. K. Vemuri, T. Büchner, J. Niebling, and J. Denzler. Functional tensor decompositions for physics-informed neural networks. *arXiv preprint arXiv:2408.13101*, 2024.
- [37] H. Wu, J. Köhler, and F. Noé. Stochastic normalizing flows. *Advances in Neural Information Processing Systems*, 33:5933–5944, 2020.
- [38] Y. Zhang, Ö. D. Akyildiz, T. Damoulas, and S. Sabanis. Nonasymptotic estimates for stochastic gradient Langevin dynamics under local conditions in nonconvex optimization. *Applied Mathematics & Optimization*, 87(2):25, 2023.