# Low Stein Discrepancy via Message-Passing Monte Carlo

**Nathan Kirk**
Illinois Institute of Technology

**T. Konstantin Rusch**
MIT

**Jakob Zech**
Heidelberg University

**Daniela Rus**
MIT

## Abstract

Message-Passing Monte Carlo (MPMC) was recently introduced as a novel low-discrepancy sampling approach leveraging tools from geometric deep learning. While originally designed for generating uniform point sets, we extend this framework to sample from general multivariate probability distributions $F$ with known probability density function. Our proposed method, Stein-Message-Passing Monte Carlo (Stein-MPMC), minimizes a kernelized Stein discrepancy, ensuring improved sample quality. Finally, we show that Stein-MPMC outperforms competing methods, such as Stein Variational Gradient Descent and (greedy) Stein Points, by achieving a lower Stein discrepancy.

## 1 Introduction

Approximating a probability distribution with a discrete set of points is a fundamental task in modern scientific computation with wide ranging applications, examples of which include uncertainty quantification, Bayesian inference, and numerical integration. All of these problems correspond to computing expectations of the form $\mathbb{E}_f(q)$ of a function $q(\boldsymbol{x})$ in $\mathbb{R}^d$ with respect to a given distribution $F$ with probability density function $f(\boldsymbol{x})$. Monte Carlo (MC) methods are a popular choice for approximating the integral by the sample mean of $q$ evaluated on a set of $N$ sample nodes $\{\mathbf{X}_i\}_{i=1}^N$ drawn IID from distribution $F$, i.e.,

$$\mathbb{E}_f(q) = \int_{\mathbb{R}^d} q(\boldsymbol{x}) f(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = \int_{\mathbb{R}^d} q(\boldsymbol{x}) \mathrm{d}F(\boldsymbol{x}) \approx \frac{1}{N}\sum_{i=1}^{N} q(\mathbf{X}_i) \tag{1}$$

for $\mathbf{X}_i \overset{\text{IID}}{\sim} F$. Provided that the variance of the integrand is bounded, the standard Monte Carlo rate of $\mathcal{O}(N^{-1/2})$ applies often necessitating a very large $N$ when high precision is required. Therefore, to obtain greater accuracy, or a faster convergence rate, one may replace the random evaluations by a carefully chosen deterministic set that better represents the distribution $F$. These so-called *low-discrepancy* points form the basis of quadrature rules that fall under the umbrella of quasi-Monte Carlo (QMC) methods Dick et al. (2013); Hickernell et al. (2025).

Assume that the function $q$ belongs to a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ of functions from $\mathbb{R}^d \to \mathbb{R}$ equipped with an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ and corresponding norm $\|\cdot\|_{\mathcal{H}}$. One can then use the Cauchy-Schwarz inequality within $\mathcal{H}$ to derive an error bound on the approximation (1) as

$$\left| \frac{1}{N}\sum_{i=1}^{N} q(\mathbf{X}_i) - \int q\, dF \right| \leq \|q\|_{\mathcal{H}}\, D_{\mathcal{H},F}\left(\{\mathbf{X}_i\}_{i=1}^N\right).$$

In the above, $D_{\mathcal{H},F}\left(\{\mathbf{X}_i\}_{i=1}^N\right)$ is referred to as the *discrepancy*, and the term $\|q\|_{\mathcal{H}}$ is a measure of variation of the integrand; see Hickernell (1998) for further details. The discrepancy term measures how closely the empirical distribution of the discrete sample point set approximates the target distribution $F$. Denote by $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ the reproducing kernel associated with the RKHS $\mathcal{H}$.

When both the integral $k_F := \int k(x, \cdot)\, dF(x) \in \mathcal{H}$ and $k_{F,F} := \int k_F \, dF$ are explicitly available, the discrepancy can be calculated directly by

$$D_{\mathcal{H},F}\left(\{\mathbf{X}_i\}_{i=1}^N\right) = \sqrt{k_{F,F} - \frac{2}{N}\sum_{i=1}^{N} k_F(\mathbf{X}_i) + \frac{1}{N^2}\sum_{i,j=1}^{N} k(\mathbf{X}_i, \mathbf{X}_j)}. \tag{2}$$

The case when $F$ is the uniform distribution is very well studied and several classical and computable measures of discrepancy exist, along with many known constructions of uniform low-discrepancy point sets and sequences; see Kuipers & Niederreiter (1974); Dick & Pillichshammer (2010); Dick et al. (2022). For a nonuniform distribution $F$, computable discrepancy measures and corresponding low-discrepancy point sets are not as widespread. Thus, hoping to exploit the existing constructions for $U[0, 1]^d$, there exist several transformations to map uniform low-discrepancy points to a nonuniform distribution $F$. For Gaussians, the Box-Muller transformation Box & Muller (1958) provides a direct and easily computable transport map coupling the uniform distribution with the target $F$. For general distributions $F$, in one dimension, the inverse CDF provides such a transport, while the Rosenblatt transformation Rosenblatt (1952) extends this approach to higher dimensions. Numerous methods have been developed to compute transport maps in practice, including normalizing flows Rezende & Mohamed (2015), neural ODEs Chen et al. (2018a), or polynomial transports Marzouk et al. (2017). Each of these approaches comes with its own challenges, mostly related to solving highly nonconvex optimization problems. Overall, it is desirable to be able to effectively generate low-discrepancy samples directly from a target distribution $F$.

When $F$ is not the uniform distribution, direct optimization of the discrepancy (equation 2) can be a difficult problem without a clear, efficient objective function to minimize. In recent years, there has emerged an active area of research on this topic using variants of a procedure derived from Stein's method Stein (1972); see Gorham & Mackey (2017); Gorham et al. (2020); Barp et al. (2019); Chen et al. (2018b); Han & Liu. (2018); Liu & Wang. (2016); Liu et al. (2016); Liu. (2017); Afzali & Muthukumarana (2023); Fisher & Oates (2024) and references therein. The papers Gorham & Mackey (2017); Chwialkowski et al. (2016); Liu et al. (2016) independently introduced the *kernel Stein discrepancy (KSD)*, one of several computable versions of the Stein discrepancy, which is used to assess the "closeness" of a sample point set to a target distribution $F$.

### 1.1 Our Contribution

In this paper, we extend the Message-Passing Monte Carlo Rusch et al. (2024) framework to minimize a kernelized Stein discrepancy, ensuring improved sample quality from general multivariate probability distributions $F$ with known probability density function. We compare Stein Discrepancy values against the point sets generated by the benchmark methods of Stein Variational Gradient Descent Liu & Wang. (2016) and Stein Points Chen et al. (2018b).

## 2 Stein Discrepancy

The work of Gorham & Mackey (2015) introduced a new family of sample quality measures, known as the Stein discrepancies, which can be used to measure the error in the approximation (1) without explicitly integrating under $F$. Stein discrepancies are derived using Stein's identity Stein (1972), a key result in probability theory that relates the expectation of a derivative-based function to properties of a target distribution. There exist several computable versions of the Stein discrepancy family, e.g., graph Stein discrepancies. However, kernel Stein discrepancies have gained most attention due to their closed-form expression involving the sum of kernel evaluations over pairs of sample points.

More formally, for a Stein operator $T_F$, the following holds

$$\int T_F[p](x)\, dF(x) = 0 \quad \forall p \in \mathcal{F},$$

where $\mathcal{F}$ is a set of functions that are sufficiently smooth. Stein's identity allows the construction of such operators $T_F$ that characterize how well a distribution matches a target. When $\mathcal{F}$ is chosen as a RKHS $\mathcal{H}$ with a reproducing kernel $k$, the image of $\mathcal{H}$ under $T_F$ is denoted as $\mathcal{H}_0 = T_F\mathcal{H}$. The KSD is then computed using the reproducing kernel $k_0$ of $\mathcal{H}_0$, defined as

$$k_0(x, x') = T_F T_F^* k(x, x'),$$

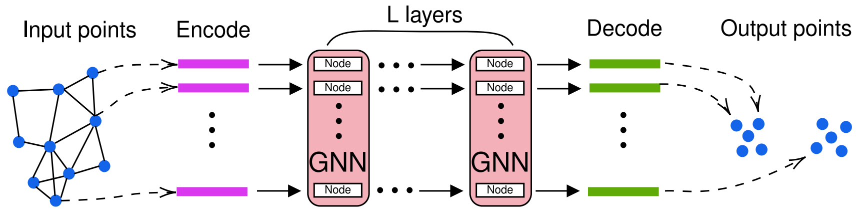


Figure 1: Schematic of the MPMC model. The points are encoded to a high dimensional representation then passed through multiple layers of a message-passing GNN where the underlying computational graph is constructed based on nearest neighbors. Finally, the node-wise output representations of the final layer of the GNN are decoded.

where $T_F^*$ is the adjoint of the Stein operator $T_F$, acting on the second argument of the kernel. A common choice for $T_F$ is the Langevin Stein operator defined by

$$T_F p(x) = \nabla \cdot (f(x)p(x))/f(x),$$

where $\nabla\cdot$ is the divergence operator, and $p$ is a vector-valued function in the RKHS $\mathcal{H}^d$. This operator leads to the Stein reproducing kernel

$$\begin{aligned} k_0(x,x') =& \nabla_x \cdot \nabla_{x'} k(x,x') + \nabla_x k(x,x') \cdot \nabla_{x'} \log f(x') \\ &+ \nabla_{x'} k(x,x') \cdot \nabla_x \log f(x) + k(x,x') \nabla_x \log f(x) \cdot \nabla_{x'} \log f(x'). \end{aligned}$$

Stein kernels possess the nice property that $k_{0,F} = \int k_0(x,\cdot)dF = 0$ and $k_{0,F,F} = \int k_{0,F} dF = 0$. Thus for some base kernel $k$, the KSD is computed from equation 2 as

$$D_{\mathcal{H}_0,F}(\{\mathbf{X}_i\}_{i=1}^N) = \sqrt{\frac{1}{N^2} \sum_{i,j=1}^{N} k_0(\mathbf{X}_i, \mathbf{X}_j)}. \tag{3}$$

## 3 Stein-Message-Passing Monte Carlo (Stein-MPMC)

Message-Passing Monte Carlo (MPMC) Rusch et al. (2024) represents a significant advancement in the field of quasi-Monte Carlo methods and general low-discrepancy sampling applications Chahine et al. (2024). MPMC leverages tools from geometric deep learning, including graph neural networks (GNNs) and a message-passing framework, to effectively learn a transformation mapping random input point set to uniform low-discrepancy points in the $d$-dimensional unit hypercube. In the original MPMC framework (see Figure 1), the target is always the uniform distribution on the $d-$dimensional hypercube and the training is guided by Warnock's formula Warnock (1972) for the $L_2$-discrepancy – a classical measure of uniformity for sample point sets in $[0,1]^d$. In the proposed Stein-MPMC model, as described below, the architecture is holistically similar, with the primary change being the Stein discrepancy based objective function.

### 3.1 Stein-MPMC Model

Our objective is to train a neural network to effectively learn a mapping to transform an initialized sample point set $\{\mathbf{X}\}_{i=1}^N$ into points $\{\hat{\mathbf{X}}\}_{i=1}^N$ that minimize the (kernel) Stein discrepancy (3) where $\mathbf{X}_i, \hat{\mathbf{X}}_i \in \mathbb{R}^d$ for all $1 \le i \le N$. The input point set will be generated randomly from target distribution $F$ where possible, i.e., $\mathbf{X}_i \overset{\text{IID}}{\sim} F$ for $1 \le i \le N$. However, in principle, our initialized training data can be taken as any reasonable set of points not judiciously chosen to be purposefully far from the target $F$.

For the model architecture, we start by constructing an undirected computational graph $G = (V, E \subseteq V \times V)$, where $V$ denotes the set of unordered nodes corresponding to the input points $\{\mathbf{X}_i\}_{i=1}^N$, and $E$ is the set of pair-wise connections between the nodes. We denote the 1-neighborhood of a node $i \in V$ as $\mathcal{N}_i = \{j \in V : (i,j) \in E\}$ and set

$$\mathcal{N}_i = \{j \in V : \|\mathbf{X}_i - \mathbf{X}_j\|_2 \le r\}$$

for a fixed radius $r \in \mathbb{R}$. That is, every node $i \in V$ is connected to every other node $j \in V$ that is within a neighborhood of radius $r$ of node $i$. This local connectivity of nodes emphasizes that the network should primarily consider near-by points when learning the transformation. The main aspect of the Stein-MPMC model is the GNN layers based on the message-passing framework. Message-passing GNNs are a family of parametric functions defined through local updates of hidden node representations. More concretely, we iteratively update node features as,

$$\mathbf{X}_i^l = \phi^l \left( \mathbf{X}_i^{l-1}, \sum_{j \in \mathcal{N}_i} \psi^l(\mathbf{X}_i^{l-1}, \mathbf{X}_j^{l-1}) \right), \quad \text{for all } l = 1, \ldots, L,$$

with $\mathbf{X}_i^l \in \mathbb{R}^{m_l}$ for all nodes $i$. Moreover, we parameterize $\phi^l, \psi^l$ as ReLU-multilayer perceptrons (MLPs), i.e., MLPs using the element-wise ReLU activation function, $\text{ReLU}(x) = \max(0, x)$, in-between layers. We further encode the initial node features by an affine transformation, $\mathbf{X}_i^0 = \mathbf{A}_{\text{enc}} \mathbf{X}_i + \mathbf{b}_{\text{enc}}$ for all $i = 1, \ldots, N$, with weight matrix $\mathbf{A}_{\text{enc}} \in \mathbb{R}^{m_0 \times d}$ and bias $\mathbf{b}_{\text{enc}} \in \mathbb{R}^{m_0}$. Finally, we decode the output of the final GNN layer by an affine transformation back into $\mathbb{R}^d$, i.e., $\hat{\mathbf{X}}_i = \mathbf{A}_{\text{dec}} \mathbf{X}_i^L + \mathbf{b}_{\text{dec}}$ for all $i = 1, \ldots, N$, with the weight matrix $\mathbf{A}_{\text{dec}} \in \mathbb{R}^{d \times m_L}$, and bias $\mathbf{b}_{\text{dec}} \in \mathbb{R}^d$.

Lastly, the training objective is selected to be the kernel Stein discrepancy (3). Reasons for this choice are two-fold; i) the kernelized version of Stein discrepancy has closed form and fast parallelizable computation of kernel evaluations of pairs of points, and ii) for carefully chosen base kernels $k$ in equation 3, there exist results that the KSD controls weak convergence to the target distribution $F$; see Chen et al. (2018b) and (Gorham & Mackey, 2017, Theorem 8).

## 4 Results

The proposed Stein-MPMC method is empirically assessed and compared against existing benchmark methods. Precisely, we illustrate across two examples of target distribution $F$ that Stein-MPMC generates sample point sets with a smaller KSD with respect to $F$. In two dimensions only, we examine a Gaussian mixture over the unbounded domain $\mathbb{R}^2$, and a distribution defined as the product of two independent Beta distributions over the unit square.

### 4.1 Experimental Detail

For the base kernel function $k$ in equation 2, we use the standard Radial Basis Function (RBF) kernel

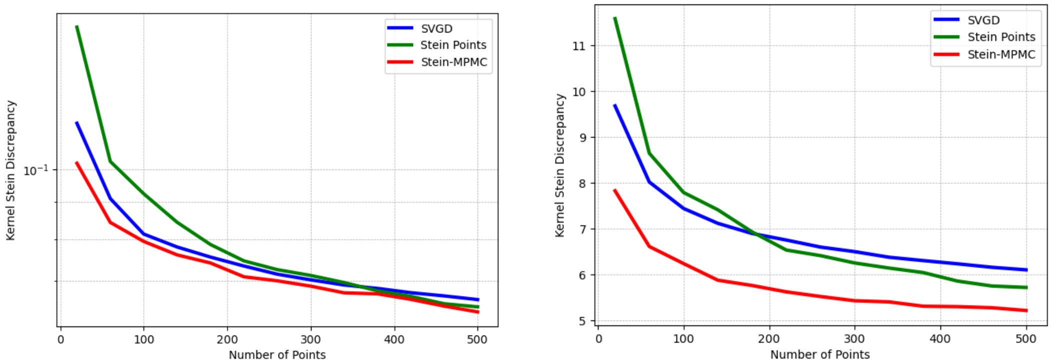
$$k(x, x') = \exp\left( -\frac{\|x - x'\|^2}{2h^2} \right).$$

where the bandwidth parameter $h$ is not tuned separately for different experiments or methods and instead, we apply the median heuristic across all experiments. Specifically, we take we take the bandwidth to be $h = \sqrt{\text{med}^2 / 2 \log(N+1)}$ where med is the median of the pairwise distances between the current sample point set. This choice is motivated to ensure a fair comparison across methods and prevent confounding effects due to kernel tuning.

We compare Stein-MPMC against two established methods:

1. Stein Variational Gradient Descent (SVGD) Liu et al. (2016) generates point sample sets from a target distribution $F$ by performing a version of gradient descent on the Kullback-Leibler (KL) divergence $\text{KL}(\cdot \| F)$. Like Stein-MPMC, SVGD is a global optimization method that updates all sample points simultaneously at each step.
2. Stein points Chen et al. (2018b) generates sequences of points from a distribution $F$ via greedy minimization of the KSD directly. Despite being a greedy sequential procedure, rather than the global methods of SVGD and Stein-MPMC, Stein points is included in the comparison due to its direct KSD optimization.

We generate samples using Stein-MPMC, SVGD, and Stein Points for values of $N$ ranging from 20 to 500 in increments of 40, recording the corresponding KSD values. For the greedy Stein Points



(a) Gaussian Mixture

(b) Beta Product

Figure 2: KSD results for our two target distributions. Stein-MPMC yields smaller KSD values for every $N = 20, 60, 100, \dots, 500$ across both distributions.

method, we track the KSD value at each instance of $N$ during a single sequential run as the sample set grows to $500$ total points.

Full experimental details including optimization methods and lists of selected hyperparameters for each method are given in Appendix A.

### 4.2 Gaussian Mixture Distribution

We first consider a Gaussian mixture model in two dimensions, which is somewhat of a standard benchmark for variational inference methods. The target distribution is a two-component Gaussian mixture

$$\frac{1}{2}\mathcal{N}(\mu_1, \Sigma_1) + \frac{1}{2}\mathcal{N}(\mu_2, \Sigma_2),$$

where $\mu_1 = (-1.5, 0), \mu_2 = (1.5, 0), \Sigma_1 = \Sigma_2 = I$. KSD values for the three methods are shown in Figure 2a, demonstrating that Stein-MPMC outperforms Stein Points and SVGD. For this Gaussian mixture example, as $N$ increases the differences between the methods become less pronounced.

### 4.3 Beta Product Distribution

We also consider a Beta-distributed target density as an example of a bounded probability distribution. The target distribution is defined as the product of two independent Beta distributions

$$X \sim \text{Beta}(\alpha_x, \beta_x), \quad Y \sim \text{Beta}(\alpha_y, \beta_y).$$

This distribution is supported on $(0, 1) \times (0, 1)$ and allows independent control over the shape of each marginal through the parameters $\alpha_x, \beta_x, \alpha_y, \beta_y$. For our experiments, we set $\alpha_x = 2, \beta_x = 4, \alpha_y = 2, \beta_y = 4$. Discrepancy values for each method for this Beta distribution are given in Figure 2b and are consistent with those of the Gaussian mixture example; Stein-MPMC outperforms the other methods with respect to KSD values for all tested instances of $N$.

## 5 Discussion

Stein-MPMC effectively minimizes kernel Stein discrepancy, outperforming SVGD and Stein Points by leveraging message-passing graph neural networks to solve the nonconvex global optimization problem. The results support the existing notion that optimizing sample point distributions for a pre-determined $N$ allows for better sample uniformity compared to sequential generation. Future work should explore its scalability to higher dimensions and the impact of adaptive kernel tuning to further enhance sample quality.

ACKNOWLEDGMENTS

The work of NK was supported by the National Science Foundation (DMS Grant No. 2316011). NK also gratefully acknowledges the Argonne Leadership Computing Facility (ALCF) for providing GPU access which supported the computational aspects of this research. This work was supported in part by the Postdoc.Mobility grant P500PT-217915 from the Swiss National Science Foundation, the Schmidt AI2050 program (grant G-22-63172), and the Department of the Air Force Artificial Intelligence Accelerator and was accomplished under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Department of the Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

## REFERENCES

Elham Afzali and Saman Muthukumarana. Gradient-free kernel conditional Stein discrepancy goodness of fit testing. *Machine Learning with Applications*, 12:100463, 2023.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework, 2019. URL `https://arxiv.org/abs/1907.10902`.

A. Barp, F.-X. Briol, A. Duncan, M. Girolami, and L. Mackey. Minimum Stein discrepancy estimators. In *In Advances in Neural Information Processing Systems*, pp. 12964–12976, 2019.

G. E. P. Box and M. E. Muller. A note on the generation of random normal deviates. *Annals of Mathematical Statistics*, 29(2):610–611, 1958.

Makram Chahine, T Konstantin Rusch, Zach J Patterson, and Daniela Rus. Improving efficiency of sampling-based motion planning via message-passing Monte Carlo. *arXiv preprint arXiv:2410.03909*, 2024.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018a. URL `https://proceedings.neurips.cc/paper_files/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf`.

W. Y. Chen, L. Mackey, J. Gorham, F.-X. Briol, and C. J. Oates. Stein points. In *ICML*, 2018b.

K. Chwialkowski, H. Strathmann, and A. Gretton. A kernel test of goodness of fit. In *JMLR: Workshop and Conference Proceedings*, 2016.

Josef Dick and Friedrich Pillichshammer. *Digital Nets and Sequences: Discrepancy Theory and Quasi–Monte Carlo Integration*. Cambridge University Press, 2010.

Josef Dick, Frances Y. Kuo, and Ian H. Sloan. High-dimensional integration: The quasi-Monte Carlo way. *Acta Numerica*, 22:133–288, 2013. doi: 10.1017/S0962492913000044.

Josef Dick, Peter Kritzer, and Friedrich Pillichshammer. *Constructions of Lattice Rules*, pp. 95–139. Springer International Publishing, Cham, 2022.

M Fisher and C Oates. Gradient-free kernel Stein discrepancy. In *NeurIPS*, 2024.

J. Gorham and L. Mackey. Measuring sample quality with Stein's method. In *Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R. (eds.), Adv. NIPS 28*, pp. 226–234. Curran Associates, Inc., 2015.

J. Gorham and L. Mackey. Measuring sample quality with kernels. In *In Proceedings of the 34th International Conference on Machine Learning*, pp. 1292–1301, 2017.

Jackson Gorham, Anant Raj, and Lester Mackey. Stochastic Stein discrepancies. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

J. Han and Q. Liu. Stein variational gradient descent without gradient. In *ICML*, 2018.

Fred Hickernell. A generalized discrepancy and quadrature error bound. *Mathematics of computation*, 67(221):299–322, 1998.

Fred J. Hickernell, Nathan Kirk, and Aleksei G. Sorokin. Quasi-monte carlo methods: What, why, and how?, 2025. URL `https://arxiv.org/abs/2502.03644`.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL `https://arxiv.org/abs/1412.6980`.

L. Kuipers and H. Niederreiter. *Uniform distribution of sequences*. Pure and Applied Mathematics. Wiley-Interscience [John Wiley & Sons], New York-London-Sydney, 1974.

Q. Liu. Stein variational gradient descent as gradient flow. In *NeurIPS*, 2017.

Q. Liu and D. Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *NeurIPS*, 2016.

Q. Liu, J. Lee, and M. Jordan. A kernelized Stein discrepancy for goodness-of-fit tests. In *International Conference on Machine Learning (ICML)*, pp. 276–284, 2016.

Youssef Marzouk, Tarek Moselhy, Matthew Parno, and Alessio Spantini. Sampling via measure transport: an introduction. In *Handbook of uncertainty quantification. Vol. 1, 2, 3*, pp. 785–825. Springer, Cham, 2017. ISBN 978-3-319-12384-4; 978-3-319-12385-1.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. 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. 1530–1538, Lille, France, 07–09 Jul 2015. PMLR. URL `https://proceedings.mlr.press/v37/rezende15.html`.

M. Rosenblatt. Remarks on a multivariat transformation. *The Annals of Mathematical Statistics*, 23 (3):470–472, 1952.

T. K. Rusch, N. Kirk, M. Bronstein, C. Lemieux, and D. Rus. Message-passing Monte Carlo: Generating low-discrepancy point sets via graph neural networks. *Proceedings of the National Academy of Sciences*, 121(40):e2409913121, 2024.

C. Stein. A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. In *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 2: Probability Theory*, pp. 583–602. University of California Press, 1972.

Tony T Warnock. Computational investigations of low-discrepancy point sets. In *Applications of number theory to numerical analysis*, pp. 319–343. Elsevier, 1972.

# A Training Details

All experiments have been run on NVIDIA DGX A100 GPUs.

## A.1 Stein-MPMC

Each model was trained for 50k epochs with the Adam optimizer Kingma & Ba (2017). Stein-MPMC hyperparameters were tuned using `Optuna` Python package Akiba et al. (2019) random search over the search spaces and distributions contained in Table 1.

| **Hyperparameter** | **Range** | **Distribution** |
|---|---|---|
| learning rate | $[10^{-4}, 10^{-2}]$ | log uniform |
| hidden size $m_0 = m_1 = \cdots = m_L$ | $\{32, 64, 128, 256\}$ | uniform |
| number of GNN layers $L$ | $\{1, 2, 3, 4, 5\}$ | uniform |
| weight decay | $[10^{-6}, 10^{-2}]$ | log uniform |

Table 1: Hyperparameter search-space and respective random distributions.

## A.2 Stein Points

Computation of the $N^{th}$ Stein point is dependent upon the already existing $N-1$ terms and requires a global optimization to find $\mathbf{X}_N \in \mathbb{R}^d$ that minimizes the kernel Stein discrepancy of the total $N$ element set, holding $\{\mathbf{X}_i\}_{i=1}^{N-1}$ fixed. In Chen et al. (2018b), several numerical optimization methods are considered to solve this $\arg\min$ problem. In our experiments, we implement the Adam optimizer with a learning rate of $0.01$, selected after testing several judiciously chosen alternatives for the learning rate.

## A.3 Stein Variational Gradient Descent

The other global optimization method considered was Stein Variational Gradient Descent introduced in Liu & Wang. (2016). SVGD was trained with the standard update rule

$$\mathbf{X}_i^{(t+1)} = \mathbf{X}_i^{(t)} + \eta \left( \frac{1}{n} \sum_{j=1}^{n} k(\mathbf{X}_i, \mathbf{X}_j) \nabla \log f(\mathbf{X}_j) + \nabla_{\mathbf{X}_j} k(\mathbf{X}_i, \mathbf{X}_j) \right)$$

where $k$ is the base kernel (taken to be the RBF kernel with median bandwidth), step size $\eta$ was fixed at $0.001$ and was run for a standard $50k$ iterations on each experiment.