Space-Filling Design for Nonlinear Models

Chang-Han Rhee[*], Enlu Zhou[**], and Peng Qiu[***]

[*]Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, 60613, USA
[**]H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 30332, USA
[***]Wallace H. Coulter Department of Biomedical Engineering, Georgia Institute of Technology and Emory University, Atlanta, GA, 30332, USA

February 28, 2023

Abstract

Performing a computer experiment can be viewed as observing a mapping between the model input parameters and the corresponding outputs predicted by the computer model to understand the input-output relationship. In view of this, experimental design for computer experiments can be thought of as devising an efficient procedure for finding configurations of design points in the input space so that their images represent the manifold parametrized by such a mapping (i.e., computer experiments) faithfully. Traditional space-filling designs aim to achieve this goal by filling the input space with design points that are as "uniform" as possible in the input space. However, the resulting design points may be non-uniform in the model output space and hence fail to unveil the input-output relationship or even become misleading in case the computer experiments are non-linear. In this paper, we propose an iterative algorithm that fills in the model output manifold uniformly—rather than the input space uniformly—so that one could obtain an efficient and reliable understanding of input-output relationship with the minimal number of design points.

1 Introduction

Due to the advances of modern computing machineries, experimentation via computer simulation has become an integral element of science and engineering. Due to its deterministic nature, however, designing computer simulation experiments requires a different approach from traditional design of physical experiments.

Let $f : \mathcal{P} \subseteq \mathbb{R}^m \to \mathbb{R}^n$ denote a function that maps the input variable $x \in \mathbb{R}^m$ to the output $f(x) \in \mathbb{R}^n$ of a computer model. The coordinates of x may consist of 1) deterministic variables that can be set by an engineer or scientist—a.k.a. control variables; 2) deterministic variables that specify the mathematical model of the computer code—a.k.a. model parameters; 3) random variables that represent the inherently uncertain components of the model—a.k.a. environmental variables; or any combination of them. In general we can write $x = (x_c, x_p, x_e)$ where $x_c \in \mathbb{R}^{m_c}$ is the set of control variables, $x_p \in \mathbb{R}^{m_p}$ is the set of model parameters, and $x_e \in \mathbb{R}^{m_e}$ is the set of environmental variables, and $m_c + m_p + m_e = m$. Typical objectives of (computer) experiments are

(O1) predicting $f(x)$ "well" for all x in the domain $\mathcal{P}$ of interest;

(O2) identifying x such that $f(x)$ meets certain criteria—i.e., finding $\mathcal{L}(C) \triangleq \{x \in \mathcal{P} : f(x) \in C\}$;

(O3) finding the extreme values (and the optima) of $f(\cdot)$.

In case the environmental variables are present so that $m_e > 0$, the above objectives are often formulated in terms of $\mu(x_c, x_p) \triangleq \mathbf{E}\, f(x_c, x_p, X_e)$—instead of $f(x_c, x_p, x_e)$—where X_e is a random vector whose distribution reflects the uncertainties in the environmental variables. In this article, we focus on the case the environmental variable is not present ($m_e = 0$), and hence, the function of our interest can be evaluated deterministically. The case where the environmental variables are present so that $\mu(x_c, x_p)$ cannot be evaluated without statistical errors will be pursued elsewhere.

In many circumstances, each computer experiment—i.e., evaluation of $f(\cdot)$ at a given point x—is computationally expensive, and hence, the design of computer experiments requires careful selection of design points $x_1, \ldots, x_k$ such that the resulting experimental data $(x_1, f(x_1)), \ldots, (x_k, f(x_k))$ is most conducive to the above goals. Since, in computer experiments, repeated observations (i.e., evaluations of f) with the same input variable produce the identical output, the design should not include duplicate points. At the same time, all portions of the experimental region should be explored. In view of these, a widely adopted principle for the design of computer experiments is to spread the design points evenly throughout the input domain $\mathcal{P}$. This type of experimental designs are called space-filling design. The exact meaning of "spreading design points evenly" is not obvious, and based on different interpretations many different space-filling designs were proposed in the literature. Detailed discussion of each space-filling design is beyond the scope of this article. We emphasize here that, while "spreading out evenly" does not necessarily coincide precisely with being distributed uniformly in $\mathcal{P}$ in probabilistic sense, there is close connection, and for the purpose of our discussion, it is enough to consider the uniform distribution as a representative example of space-filling design. For detailed discussion of traditional space-filling design, see for example, Santner et al. (2013) and the references therein.

In this paper, we argue that in case the computer model is nonlinear, a design scheme that spreads the output $\{f(x_1), \ldots, f(x_k)\}$ evenly in the output space (rather than spreading $\{x_1, \ldots, x_k\}$ evenly in the input space) can be more efficient for achieving the above mentioned objectives. Spreading the output evenly in the output space corresponds to sampling uniformly from the manifold $\mathcal{M} \triangleq f(\mathcal{P})$. We will make what we mean by this clear in rigorous mathematical terms in

Section 2, but here we first explain our motivation and objective at an intuitive level through an illustrative example. Consider a mapping $f : \mathcal{P} \subseteq \mathbb{R}^2 \to \mathbb{R}^3$

$$f(\theta_1, \theta_2) \triangleq \begin{pmatrix} e^{-\theta_1 t_1} + e^{-\theta_2 t_1} \\ e^{-\theta_1 t_2} + e^{-\theta_2 t_2} \\ e^{-\theta_1 t_3} + e^{-\theta_2 t_3} \end{pmatrix}$$

on $\mathcal{P} = [0, 100]^2$ and the associated manifold

$$\mathcal{M} = \{(e^{-\theta_1 t_1} + e^{-\theta_2 t_1}, e^{-\theta_1 t_2} + e^{-\theta_2 t_2}, e^{-\theta_1 t_3} + e^{-\theta_2 t_3}) : \theta_1, \theta_2 \in [0, 100]\} \tag{1.1}$$

where $t_1 = 1$, $t_2 = 2$, $t_3 = 4$. Although this example is given in a closed-form formula for the purpose of illustration, a typical situation we address in this paper is when the evaluation of f is only possible through an expensive black box simulation. Here we point out that while this example is schematic, it captures important aspects of more complex nonlinear behaviors that arise in real-life examples such as Example 4. Think of f as a model describing the dynamics of the system so that $f(\theta_1, \theta_2)$ is the model output given the input (θ_1, θ_2). $\mathcal{M}$ then can be viewed as all of the possible behaviors of the system that correspond to the inputs in $\mathcal{P}$. Traditional space-filling designs carefully construct design points $\{x_1, \ldots, x_k\}$ in such a way that the input space $\mathcal{P}$ is "well covered" by $\{x_1, \ldots, x_k\}$. While such a strategy can be a powerful means to understanding the input-output relationship of the mapping f, it should be noted that if f is parametrized in a highly nonlinear way so that the vast majority of the input space $\mathcal{P}$ is mapped into a small part of the manifold $\mathcal{M}$, then $\{x_1, \ldots, x_k\}$ that are spread uniformly in $\mathcal{P}$ can lead to not only very inefficient computational procedures but also dangerously misleading observations. The manifold $\mathcal{M}$ in (1.1) illustrates this point. Figure 1 displays the samples generated on $\mathcal{M}$ in two different ways. The left plot displays $5,000$ samples $f(x_1), \ldots, f(x_{5,000})$ where x_i-s are generated uniformly on $\mathcal{P}$, whereas the right plot displays $5,000$ samples $f(x'_1), \ldots, f(x'_{5,000})$ where x'_i-s are generated so that $f(x'_i)$-s are uniformly distributed on $\mathcal{M}$. One can see that most of the x_i-s are mapped into a small fraction of $\mathcal{M}$ in the left plot, and hence, $\{f(x_1), \ldots, f(x_k)\}$ fail to reflect the actual geometry of $\mathcal{M}$. Such an outcome translates to a poor performance in terms of the experimental goals (O1)-(O3). For example, if one tries to predict the output of an unexamined point x_0 based on the uniform samples in the input space, the prediction can only fall somewhere near the limited output points discovered in the left plot of Figure 1, whereas, in fact, the output of x_0 can fall anywhere on the manifold shown in the right plot. With the right plot, however, one can yield much more accurate predictions for unexamined points even with a simple interpolation. Likewise, (O2) and (O3) can also be achieved much more efficiently by spreading design points evenly in the output space rather than the input space. This clearly illustrates the potential danger in drawing conclusions from the observations based on blindly choosing uniform configurations in the input space without considering the behavior of the model f. An obvious alternative is to consider the space-filling design in output space. A more rigorous and comprehensive discussion of the merits and demerits of space-filling design in output space compared to that in input space in terms of the efficiency and prediction capability will be left as a future research direction. In this paper, we instead focus on the computational aspect of constructing design points that are spread out evenly (or according to a given target distribution) in output space. More specifically, we propose computational procedures for generating configurations (i.e., design points) $\{x_1, \ldots, x_k\}$ on $\mathcal{P}$ so that the resulting configuration $\{f(x_1), \ldots, f(x_k)\}$ on $\mathcal{M}$ is uniformly distributed (or according to other desired distributions) as in the right plot of Figure 1. Therefore, our target distribution is the distribution on $\mathcal{P}$ that corresponds to the uniform distribution on $\mathcal{M}$. The idea is to start with a random configuration, and then shift the configuration towards the target distribution by alternating between resampling in the model output space and perturbation in the input space. In the resampling step, the algorithm resamples the design points in such a way that the points that belong to a densely populated region of $\mathcal{M}$ are less likely to be resampled, while the points that belong to the sparse region of $\mathcal{M}$ are more likely to be resampled so that the resulting samples are populated more uniformly. While this step pushes the empirical distribution of the design points toward the target distribution, it cannot be repeated to push the design points further toward the target distribution since no new points in $\mathcal{M}$ are discovered while some are discarded. To discover

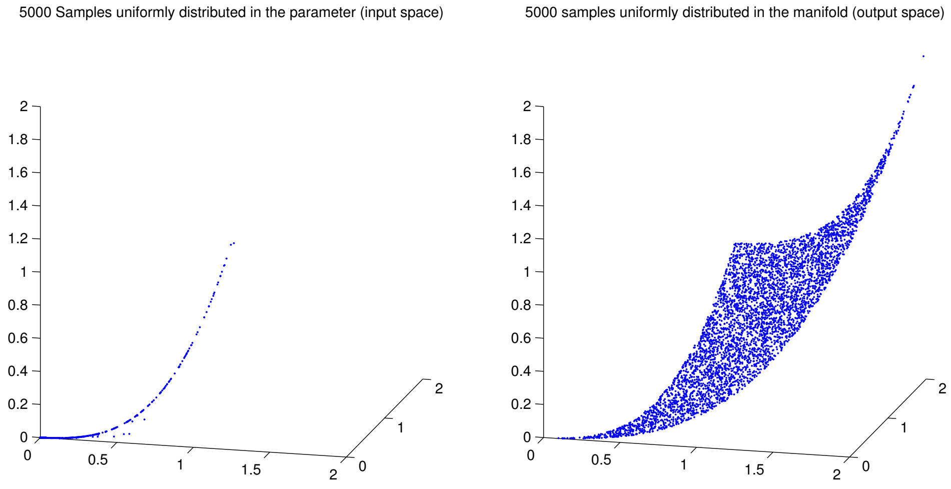

Figure 1: $5,000$ samples generated uniformly on the input space $\mathcal{P}$ (left plot) vs. uniformly on the manifold $\mathcal{M}$ (right plot). Observations based on the left plot may lead to incorrect inferences. For many purposes, more reliable and informative configurations for such nonlinear models would be uniform (or other desired) distribution on $\mathcal{M}$ rather than $\mathcal{P}$.

new regions of $\mathcal{M}$, the resampling step is followed by the perturbation step, where the resampled points are perturbed in such a way that the distribution of the perturbed points stay close to the original distribution. The two steps are repeated until the configuration is sufficiently uniform on $\mathcal{M}$. We study the consistency and the convergence of the proposed algorithm in Kantorovich-Rubinstein distance. In particular, we prove that the empirical distribution of the samples generated in each iteration converges to the target distribution. It should be noted that while this implies that the empirical distribution of the entire set of samples generated throughout all the past iterations will also converge to the target distribution, the samples generated in the early iterations are likely to be significantly different from the ones in the later iterations (and from the target distribution). If our objective is solely generating samples uniformly on $\mathcal{M}$, those early samples can be discarded. However, with the objectives (O1)-(O3) in the computer experiment contexts, it is often better to keep the early samples as well.

Finally, note that our setting is different from the setting where algorithms generate uniform samples on manifolds based on random walks, such as hit-and-run Boneh and Golan (1979); Smith (1984), shake-and-bake Boender et al. (1991), stochastic billiard Dieker and Vempala (2015), and geodesic-walks Lee and Vempala (2016). Such algorithms often require that the manifold is a convex set or the boundary of a convex set, and the manifold is specified by a set of constraints (eg. a polytope given by a system of linear inequalities), and hence, it is easy to tell whether a given point in $\mathbb{R}^n$ (the ambient space in which the manifold is embedded) belongs to the manifold or not. In contrast, in our setting, the manifold is not necessarily a convex set or the boundary of a convex set, and more importantly, we cannot directly answer whether a given point in the output space $\mathbb{R}^n$ belongs to the manifold $\mathcal{M}$ or not.

The rest of the paper is organized as follows. Section 2 formulates the objectives of this paper in rigorous mathematical formulation and presents our main algorithm. Section 3 analyzes the consistency and the convergence of the proposed algorithm. The proofs of the main result in Section 3 is provided in Appendix A. Section 4 examines the numerical behavior of our algorithm with a few illustrating examples including a systems biology model for enzymatic reaction networks.

2 Problem Formulation and Algorithm Description

Section 2.1 defines the preliminary mathematical notions required and state the objectives of this paper, and Section 2.2 proposes the algorithm that achieves the objectives.

2.1 Problem Formulation

Let $f : \mathcal{P} \subset \mathbb{R}^m \to \mathbb{R}^n$ be a one-to-one mapping on a hypercube $\mathcal{P} \triangleq \prod_{i=1}^{m}[x^i_{\text{min}}, x^i_{\text{max}}] \subseteq \mathbb{R}^m$ with sufficient regularity so that $\mathcal{M} \triangleq f(\mathcal{P})$ is the associated m-dimensional manifold embedded in $\mathbb{R}^n$. Note that requiring f to be one-to-one is a rather strong assumption. In the general cases where there is no guarantee that f is one-to-one, we can consider an augmented mapping $\tilde{f}^\epsilon : \mathcal{P} \to \mathbb{R}^{m+n}$ such that $\tilde{f}^\epsilon(x) \triangleq (f(x), \epsilon x)$ and apply the proposed algorithm to $\tilde{f}^\epsilon$ instead of f for small $\epsilon > 0$ to obtain roughly "uniform" samples in the output space. The exact mathematical characterization of the limit $\epsilon \to 0$ and the choice of ϵ for practical computation will be left to future study. For simplicity, we focus on the case f is one-to-one in this paper. We assume that f is not analytically tractable but can be evaluated by a simulation code at arbitrary points in $\mathcal{P}$; however, such evaluation is computationally expensive. Our goal is to find design points $\{x_1, \ldots, x_k\} \subseteq \mathcal{P}$, such that $\{f(x_1), \ldots, f(x_k)\}$ are distributed uniformly (or according to a desired distribution) on the manifold $\mathcal{M}$. A natural notion of uniform distribution on a manifold can be stated in terms of the Hausdorff measure. Recall that the m-dimensional Hausdorff measure (embedded in $\mathbb{R}^n$, $n \geq m$) is defined as

$$\mathcal{H}^m(B) = \lim_{\delta \to 0} \inf_{\substack{B \subseteq \cup S_i, \\ \operatorname{diam}(S_i) \leq \delta}} \sum \Gamma_m \left(\frac{\operatorname{diam}(S_i)}{2} \right)^m \tag{2.1}$$

where the infimum is over all countable coverings $\{S_i \subseteq \mathbb{R}^n : i \in I\}$ of B, $\operatorname{diam}(S_i) \triangleq \sup\{|x - y| : x, y \in S_i\}$, and Γ_m is the volume of the m-dimensional unit ball. Note that $\mathcal{H}^m$ is a natural generalization of Lebesgue measure, and if $m = n$, $\mathcal{H}^m$ coincides with the m-dimensional Lebesgue measure; see, for example, Federer (1996) for more details.

As Diaconis et al. (2013) points out, the area formula (see, for example, Section 3.2.5 of Federer 1996) in geometric measure theory dictates how one should sample from a given density with respect to the Hausdorff measure.

Proposition 1. *(Area Formula, Federer 1996) Let $m \leq n$. If $f : \mathbb{R}^m \to \mathbb{R}^n$ is Lipschitz and one-to-one, for any measurable A and measurable $g : \mathbb{R}^n \to \mathbb{R}$,*

$$\int_A g(f(x)) J_m f(x) \lambda^m(dx) = \int_{f(A)} g(y)\ \mathcal{H}^m(dy), \tag{2.2}$$

where λ^m denotes the m-dimensional Lebesgue measure and $J_m f$ is the m-dimensional Jacobian of f. In our context where $m \leq n$, the m-dimensional Jacobian is equal to

$$J_m f(x) = \sqrt{\det\bigl(Df(x)^T Df(x)\bigr)},$$

where $Df(x)$ is the differential of f at x:

$$Df = \begin{pmatrix} \frac{\partial}{\partial x_1} f_1 & \frac{\partial}{\partial x_2} f_1 & \cdots & \frac{\partial}{\partial x_m} f_1 \\ \frac{\partial}{\partial x_1} f_2 & \frac{\partial}{\partial x_2} f_2 & \cdots & \frac{\partial}{\partial x_m} f_2 \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial}{\partial x_1} f_n & \frac{\partial}{\partial x_2} f_n & \cdots & \frac{\partial}{\partial x_m} f_n \end{pmatrix}.$$

The area formula essentially transforms a distribution on the manifold (with respect to the Hausdorff measure $\mathcal{H}^m$) to the corresponding distribution on the input space $\mathcal{P}$ associated with the mapping f. The area formula implies that our target distribution on $\mathcal{P}$ is $J_m f$ (with respect to the Lebesgue measure), if we intend to generate uniform samples on the output manifold. More generally, we assume that $\eta(\cdot)$ is absolutely continuous with respect to the Hausdorff measure $\mathcal{H}^m$ on $\mathcal{M}$, and the density (i.e., Radon-Nikodym derivative) is μ, which is known up to a multiplicative constant. In view of the area formula (2.2), to generate samples from η, one can generate samples $x_1, x_2, \ldots$ in the input space from the density proportional to $\mu \circ f \cdot J_m f$ and then apply f to $x_1, x_2, \ldots$ to obtain the samples $f(x_1), f(x_2), \ldots$ from the desired density on the manifold. To see at an intuitive level why such a procedure generates samples from the desired distribution, pick $g(x) = \mu(x) I_B(x)$ for a given set B. Then from (2.2), $P(X \in f^{-1}(B)) = \int \mu \circ f(x) J_m f(x) I_{f^{-1}(B)}(x) dx = \int \mu(y) I_B(y) \mathcal{H}^m(dy) = P(Y \in B)$, where Y is an $\mathcal{M}$-valued random variable with the density μ, and X is an $\mathcal{P}$-valued random variable with density ξ. Since B was chosen arbitrarily, we see that $f(X)$ and Y have the same distribution. That is, if we sample X in $\mathcal{P}$ from the density ξ (w.r.t. the Lebesgue measure), then $f(X)$ is a random variable in $\mathcal{M}$ with density μ (w.r.t. the Hausdorff measure $\mathcal{H}^m$). We denote our target distribution (in the input space $\mathcal{P}$) with ξ:

$$\frac{d\xi}{d\lambda^m}(x) = \frac{\mu \circ f(x) \cdot J_m f(x)}{\int_{\mathcal{P}} \mu \circ f(x) \cdot J_m f(x) \ \lambda^m(dx)}. \tag{2.3}$$

We will call both η and ξ target distribution, and μ and $\mu \circ f \cdot J_m f$ target density when the context is clear.

Sampling from a given density with respect to the Lebesgue measure is a classical topic that has been addressed by many traditional methods such as inversion, acceptance-rejection, and Markov chain Monte Carlo (MCMC); in particular, Markov chain Monte Carlo algorithms such as Metropolis-Hastings provide powerful means to sample from analytically intractable densities; see for example Asmussen and Glynn (2007); Liu (2008); Robert (2004). However, it should be noted that our goal is different from the context where MCMC methods are typically deployed. MCMC algorithms produce samples that conform with the target distribution by rejecting many proposals that do not conform with the target distribution. Deciding whether or not to reject the proposal requires computation of likelihoods, which corresponds to performing computer experiments in our context.

For example, we can easily obtain nearly uniform samples on $\mathcal{M}$ in (1.1) from a simple Metropolis-Hastings (MH) algorithm as shown in the left plot of Figure 2. However, to obtain 3,733 near-uniform points in the left plot, the MH algorithm made 10,000 proposals and rejected 6,267 of them, for each of which $J_m f$ had to be evaluated to decide whether or not to reject: see Appendix C for details of this experiment. The right plot shows all the points at which we had to evaluate $J_m f$ to obtain the uniform points on the left plot. Note that this non-uniformity does not disappear as we increase the run length and has little to do with the initial transient problem. Of course, the efficiency of an MCMC algorithm in our context depends critically on the choice of the proposal kernel. Our algorithm can potentially provide a principled way of constructing an efficient proposal for MCMC algorithms and such a combination might produce an algorithm with improved efficiency for certain tasks. However, we do not pursue such a hybrid strategy in this paper and leave it as a future research direction.

It should also be noted that there are recent developments on "rejection-free" algorithms in the MCMC literature (Rosenthal et al., 2019; Bouchard-Côté et al., 2018). Despite the nomenclature, however, such algorithms are not designed to remove the rejection scheme from the sampling procedure; rather, the objectives are to improve the mixing properties of the MCMC algorithms using non-reversible processes and parallel computing structures, and the likelihood should still be evaluated at the points that are not included in the final output.

It is also worth mentioning that the proposed procedures bears resemblance to the sequential Monte Carlo method for Bayesian state-space models, where at every stage of the sequential problem the posterior distribution would be updated by resampling the particles, and then the particles would be perturbed to combat degeneracy. For example, the resample-move algorithm proposed in Gilks and Berzuini (2001) follows this framework. While their resample-move algorithm is typically used in filtering contexts where the goal is to study multiple target distributions evolving over time, the most general framework proposed in Section 3.2 of Gilks and Berzuini (2001) applies to more general contexts. However, if the algorithm is applied to our problem where the target distribution stays the same throughout the iterations, their resampling step reduces to an unweighted bootstrap sampling since the weights become all 1. Combined with the move step, which perturbs the particles according to a transition kernel whose invariant distribution coincides with the target distribution, the whole resample-move algorithm becomes very similar to the standard MCMC algorithms, and hence, faces the same challenge—construction of an efficient proposal—in our context.

Among the existing sampling techniques, what comes closest to our spirit is non-parametric importance sampling: Zhang (1996); Givens and Raftery (1996); Kim et al. (2000); Zlochin and Baram (2002). In fact, if the evaluation of the derivative of f is also available in addition to the evaluation of f itself, our algorithm can be simplified to a version which can be seen as a variant of non-parametric importance sampling algorithms. However, previous studies of such algorithms have

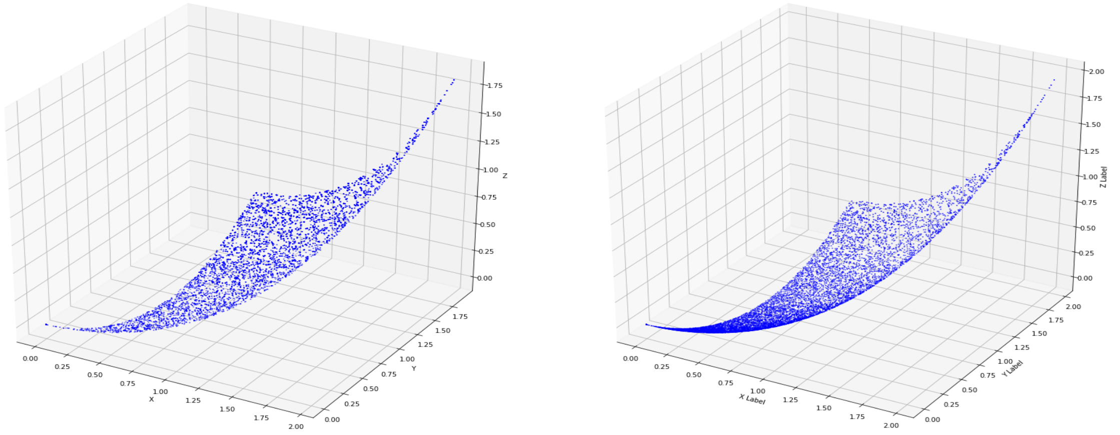

Figure 2: Left plot shows the 3733 samples generated by a simple Metropolis-Hastings (MH) algorithm. MH does produce points uniformly distributed over $\mathcal{M}$, but to generate these points, MH algorithm made 10,000 proposals shown on the right plot. These proposals are far from uniform distribution, and most of the rejected proposals are concentrated near $(0,0,0) \in \mathcal{M}$. Since each proposal requires the evaluation of the mapping f for one to decide whether or not to accept the proposal, these types of approaches (i.e., sampling methods based on acceptance-rejection schemes that require evaluation of f for computing the acceptance probability) do not serve our the purpose.

been focused on computing a single test function by approximating the zero-variance importance sampling measure with kernel density proposal. In view of our purpose, a distance between the target measure and the empirical measure of the samples produced by the algorithm would be a more proper perfomance measure. We analyze the convergence of our algorithm with respect to the Kantorovich-Rubinstein distance (which is also known as the Wasserstein distance of order 1). To the best of the authors' knowledge, the convergence bound we establish for our algorithm in this paper is the first convergence analysis of non-parametric importance sampling type algorithms in terms of Kantorovich-Rubinstein distance.

We conclude this section with a brief review of the Kantorovich-Rubinstein distance. The Kantorovich-Rubinstein distance between two probability distributions is the L^1 distance between the optimal coupling of the two associated random variables whose marginal distributions coincide with the two probability distributions. That is, for probability measures μ and ν on $\mathcal{P}$, the Kantorovich-Rubinstein distance $\mathcal{W}_1$ is defined as

$$\mathcal{W}_1(\mu,\nu) \triangleq \inf_{\pi \in M(\mu,\nu)} \int_{\mathcal{P}\times\mathcal{P}} \|x-y\|_1 d\pi(x,y)$$

where $M(\mu,\nu)$ denotes the set of all joint probability measures on $\mathcal{P}\times\mathcal{P}$ with marginals μ and ν

respectively. Obviously, this is equivalent to

$$\mathcal{W}_1(\mu, \nu) = \inf_{X,Y} \mathbf{E} \, \|X - Y\|_1$$

where the infimum is taken over all coupling of μ and ν. The following dual formula will be useful in the analysis of the modulus of continuity of the resampling step: for any given $x^0 \in \mathcal{P}$,

$$\mathcal{W}_1(\mu, \nu) = \sup \left\{ \int_{\mathcal{P}} f(x)\mu(dx) - \int_{\mathcal{P}} f(x)\nu(dx) \;\middle|\; f : \mathcal{P} \to \mathbb{R},\ \|f\|_{\text{Lip}} \leq 1,\ f(x^0) = 0 \right\} \tag{2.4}$$

where $\|f\|_{\text{Lip}}$ denotes the minimal Lipschitz constant of f. The Kantorovich-Rubinstein distance metrizes the weak convergence, that is, convergence in $\mathcal{W}_1$ implies weak convergence. See, for example, Chapter 6 of Villani (2008) for more details.

2.2 Algorithm Description

In this section, we propose an algorithm that generates a sequence of design points whose empirical distribution converges to the target distribution in Kantorovich-Rubinstein distance.

The main idea of our algorithm is to start with arbitrarily distributed samples and then repeat iterations consisting of a resampling step and a perturbation step so that the empirical distribution of the generated samples becomes closer and closer to the target distribution. The resampling step is designed to shift the current empirical distribution toward the target distribution by eliminating the samples in concentrated regions, and duplicating the samples in sparse regions; the perturbation step is designed to force the algorithm to explore new areas of the manifold while still respecting the information obtained through the previous iterations. More specifically, the algorithm works as follows. At iteration 0, one starts with N (arbitrarily chosen) points $x_1, \ldots, x_N \in \mathcal{P}$ and their images $y_1, \ldots, y_N \in \mathcal{M}$ where $y_i \triangleq f(x_i)$ for $i = 1, \ldots, N$. Let $B(y_i; r)$ denote the n-dimensional ball with radius r centered at y_i, and $\hat{r}(y_i)$ denote the k^{th} nearest neighborhood (k-NN) distance from y_i. That is, $\hat{r}(y_i) = \hat{r} \circ f(x_i) \triangleq \inf\{r > 0 : \#\{j : y_j \in B(y_i; r)\} \geq k\}$ where $\#A$ denotes the cardinality of set A. For each $j > 0$, the j^{th} iteration consists of a resampling step and a perturbation step. In the resampling step, one computes the resampling weights G_i as follows:

$$G_i \triangleq \frac{\big(\hat{r}^m \circ f(x_i)\big) \cdot \big(\mu \circ f(x_i)\big)}{\sum_{l=1}^N \big(\hat{r}^m \circ f(x_l)\big) \cdot \big(\mu \circ f(x_l)\big)}, \qquad \forall i = 1, \ldots, N \tag{2.5}$$

where $\hat{r}^m(\cdot)$ simply denotes the m^{th} power of $\hat{r}(\cdot)$—i.e., $\hat{r}^m(y) = (\hat{r}(y))^m$ for $y \in \mathbb{R}^n$. That is, we will resample according to a probability measure proportional to the values of $\hat{r}^m \circ f \cdot \mu \circ f$ at $x_1, \ldots, x_N$.

Then, the algorithm generates independent and identically distributed (iid) samples $x'_1, \ldots, x'_N$

in such a way that $\mathbf{P}(x_i' = x_j) = G_j$ for each $i, j = 1, \ldots, N$. For the perturbation step, pick a scaled kernel $\tilde{\zeta}_h(\cdot; y)$ centered at y with bandwidth $h > 0$. The constants q and b regularize the perturbation density so that the density is bounded away from 0 and ∞, while h is the perturbation bandwidth. The choice of these parameters and the precise construction of $\tilde{\zeta}_h$ will be discussed further in Section 3 and Section 4. One starts the perturbation step with the samples $x_1', \ldots, x_N'$ generated in the previous resampling step and constructs a smoothed and regularized density

$$\frac{\min\left\{b,\ q/\lambda^m(\mathcal{P}) + (1-q)\frac{1}{N}\sum_{i=1}^N \tilde{\zeta}_h(x; x_i')\right\}}{\int_A \min\left\{b,\ q/\lambda^m(\mathcal{P}) + (1-q)\frac{1}{N}\sum_{i=1}^N \tilde{\zeta}_h(s; x_i')\right\} ds} \tag{2.6}$$

for $b \gg 1$ and $q \in (0, 1)$. This is a density obtained by first mixing the uniform distribution on $\mathcal{P}$ with the mixing coefficient q and the empirical distribution of $x_1', \ldots, x_N'$ smoothed by $\tilde{\zeta}_h$, and then truncating the density of the mixture at b. The uniform distribution in the mixture provides a global exploration of the input space $\mathcal{P}$, while the smoothed kernel provides a local perturbation. To generate a sample x_j from this density (2.6), the algorithm first generates a proposal x^* uniformly (on $\mathcal{P}$) with probability q, and according to $\frac{1}{N}\sum_{i=1}^N \tilde{\zeta}_h(\cdot; x_i')$ with probability $1 - q$. Set $a = q/\lambda^m(\mathcal{P}) + (1-q)\frac{1}{N}\sum_{i=1}^N \tilde{\zeta}_h(x^*; x_i')$. Accept the proposal x^* (i.e., set $x_j \leftarrow x^*$) with probability $\min\{a, b\}/a$. If not accepted, generate another proposal and repeat this acceptance/rejection step until a proposal is accepted. When $x_1, \ldots, x_N$ are all generated from this procedure, one moves on to the resampling step of the next iteration. We note that although we use acceptance/rejection scheme here to generate samples from (2.6), the density does not explicitly involve quantities associated with f, and hence, there is no extra computation of f or $J_m f$ in this step. The whole procedure described so far is summarized in Algorithm 1.

Throughout the rest of this section, we provide some intuition behind the design and analysis of Algorithm 1, in particular, the resampling formula (2.5) and propose a simplified version—Algorithm 2—in case the derivative of f can be evaluated in addition to the value of f itself. Recall first that, in general, one can shift a given empirical distribution of iid samples toward a target distribution by reweighting the empirical distribution w.r.t. the likelihood ratio between the current distribution and the target distribution, i.e., imporatance sampling. In our context, $1/\hat{r}^m$ is approximately proportional to the density p_Y of the current samples $y_1, \ldots, y_N$, and hence, the resampling formula (2.5) assigns probability mass approximately proportional to the likelihood ratio μ/p_Y between the target density μ (on $\mathcal{P}$) and the density of the current samples p_Y. In view of this, the resulting empirical distribution of the samples after the resampling step should be closer to the target distribution.

To be more specific, let δ_y denote a unit point measure concentrated on y and suppose that $x_1, \ldots, x_N$ are the iid samples generated by the perturbation step in the previous iteration. We argue that the weighted empirical measure $\hat{\eta} \triangleq \frac{1}{N}\sum_{i=1}^N G_i \delta_{f(x_i)}$ resulting from the resampling

Algorithm 1 Space-Filling Algorithm (without Derivative)

```
Generate $N$ samples $x_1, \cdots, x_N \in \mathcal{P}$ from an initial distribution $p_0$;
while $\hat{\eta}$ changes notably do
    $\{x'_1, \cdots, x'_N\} \leftarrow$ Resample($\{x_1, \cdots, x_N\}$);
    $\{x_1, \cdots, x_N\} \leftarrow$ Perturb($\{x'_1, \cdots, x'_N\}$);
    $\hat{\eta} \leftarrow \frac{1}{N} \sum_{i=1}^{N} \delta_{f(x_i)}$;
end while
return $\hat{\eta}$

function Resample($\{x_1, \cdots, x_N\}$)
    $y_i \leftarrow f(x_i)$,                                          $i = 1, \ldots, N$;
    $\hat{r}_i \leftarrow k$-NN distance from $y_i$,                  $i = 1, \ldots, N$;
    $G_i \leftarrow \hat{r}_i^m \cdot \mu(y_i) / (\sum_{l=1}^{n} \hat{r}_l^m \cdot \mu(y_l))$,   $i = 1, \ldots, N$;
    for $i = 1 : N$ do
        Sample $x'_i$ so that $\mathbf{P}(x'_i = x_l) = G_l$, $\forall l = 1, \ldots, N$;
    end for
    return $\{x'_1, \cdots, x'_N\}$;
end function

function Perturb($\{x'_1, \cdots, x'_N\}$)
    for $i = 1 : N$ do
        while $x^*$ is not accepted do
            Draw $x^*$ from the density $q/\lambda^m(\mathcal{P}) + (1-q)\frac{1}{n}\sum_{l=1}^{n} \tilde{\zeta}_h(x; x'_l)$;
            $a \leftarrow q/\lambda^m(\mathcal{P}) + (1-q)\frac{1}{n}\sum_{l=1}^{n} \tilde{\zeta}_h(x^*; x'_l)$;
            Accept $x^*$ and set $x_i = x^*$ with probability $\frac{\min\{a,b\}}{a}$;
        end while
    end for
    return $\{x_1, \cdots, x_N\}$;
end function
```

formula (2.5) is an approximation of the target measure η. To see why, we view $\hat{\eta}$ and η in terms of Boltzmann-Gibbs transformation. Recall that the Boltzmann-Gibbs transformation $\Psi_G(\eta)$ of a measure η on $\mathcal{M}$ w.r.t. a potential $G : \mathcal{M} \to \mathbb{R}_+$ is defined as a measure such that

$$\Psi_G(\eta)(dy) = \frac{G(y)\eta(dy)}{\int G(y)\eta(dx)}.$$

Recall also that we denoted the volume of the m-dimensional unit ball with Γ_m. Let $\hat{p}_Y(y) \triangleq (k/N)/(\Gamma_m \hat{r}^m(y))$ and note that this can be understood as the estimated density of y_i's via k-NN. Set a potential $\hat{G}_Y$ as

$$\hat{G}_Y(y) \triangleq \mu(y)/\hat{p}_Y(y) = \mu(y)\Gamma_m \hat{r}^m(y)/(k/N)$$

and denote the empirical distribution of y_i's with $\hat{\eta}_Y = \frac{1}{N}\sum_{i=1}^N \delta_{y_i}$. With these notations,

$$\hat{\eta} = \Psi_{\hat{G}_Y}(\hat{\eta}_Y). \tag{2.7}$$

Let μ_Y and p_Y denote the distribution and density of y_i's and $r^m_{k,N}(z)$ denote the m^{th} power of the diameter of the ball that contains the k/N fraction of y_i's distribution. That is, for each $z \in \mathcal{M}$, $r_{k,N}(z)$ satisfies

$$\int_{B(z;r_{k,N}(z))} p_Y(y)\mathcal{H}^m(dy) = k/N.$$

Set the potential G_Y to be

$$G_Y(y) \triangleq \lim_{N,k\to\infty,\ k/N\to 0} \mu(y)\Gamma_m r^m_{k,N}(y)/(k/N) = \mu(y)/p_Y(y),$$

for k and N that increase at the rates for which the k-NN density estimation is consistent. Note that $\hat{r}$ approximates $r_{k,N}$. With these notations, we see that η can be rewritten as

$$\eta = \Psi_{G_Y}(\eta_Y). \tag{2.8}$$

In view of (2.7) and (2.8), we can expect that the difference between η and $\hat{\eta}$ will be small if $\hat{\eta}_Y$ is a good approximation of η_Y, and $\hat{G}_Y$ is a good approximation of G_Y, which explains at an intuitive level why the resampling formula (2.5) works. The crux of our algorithm analyses in Section 3 and Appendix A, B are to bound this difference and show that the iterative application of the resampling step and the perturbation step is stable and convergent.

Finally, recall that our target measure ξ on the input space $\mathcal{P} \subset \mathbb{R}^m$ is the probability measure with the density proportional to $\mu \circ f \cdot J_m f$; see (2.3). It should also be noted that $p_X(x) = J_m f(x) \cdot p_Y \circ f(x)$, and hence, $(\hat{r}^m \circ f \cdot \mu \circ f)$ in (2.5) can be regarded as an approximation of $(\mu \circ f \cdot J_m f \cdot \iota)$—up to a multiplicative constant—where $\iota \triangleq 1/p_X$ is the reciprocal of the density from which x_i-s are sampled from. In view of this, if one can readily evaluate $J_m f$, a simplified version of Algorithm 1 can be implemented by replacing $(\hat{r}^m \circ f \cdot \mu \circ f)$ in (1) with $(\mu \circ f \cdot J_m f \cdot \iota)$. In this case, the perturbation step can also be simplified; it is not necessary to truncate the sampling density in the perturbation step at level b. Such a simplified version of the algorithm is summarized in Algorithm 2.

Algorithm 2 Space-Filling Algorithm (with Derivative)

```
Generate $N$ samples $x_1, \cdots, x_N \in \mathcal{P}$ from an initial distribution $p_0$;
while $\hat{\eta}$ changes notably do
    $\{x'_1, \cdots, x'_N\} \leftarrow$ Resample$(\{x_1, \cdots, x_N\})$;
    $\{x_1, \cdots, x_N\} \leftarrow$ Perturb$(\{x'_1, \cdots, x'_N\})$;
    $\hat{\eta} \leftarrow \frac{1}{N}\sum_{i=1}^{N} \delta_{f(x_i)}$;
end while
return $\hat{\eta}$

function Resample$(\{x_1, \cdots, x_N\})$
    $\mu_i \leftarrow \mu \circ f(x_i)$                                                        $i = 1, \ldots, N$;
    $J_i \leftarrow J_m f(x_i)$                                                           $i = 1, \ldots, N$;
    $\iota_i \leftarrow \big(q/\lambda^m(\mathcal{P}) + (1-q)\frac{1}{N}\sum_{l=1}^{N} \tilde{\zeta}_h(x_i; x'_l)\big)^{-1}$     $i = 1, \ldots, N$;
    $G_i \leftarrow \mu_i \cdot J_i \cdot \iota_i / (\sum_{l=1}^{n} \mu_i \cdot J_i \cdot \iota_l)$                    $i = 1, \ldots, N$;
    for $i = 1 : N$ do
        Sample $x'_i$ so that $\mathbf{P}(x'_i = x_l) = G_l$, $\forall l = 1, \ldots, N$;
    end for
    return $\{x'_1, \cdots, x'_N\}$;
end function

function Perturb$(\{x'_1, \cdots, x'_N\})$
    for $i = 1 : N$ do
        Draw $x^*$ from the density $q/\lambda^m(\mathcal{P}) + (1-q)\frac{1}{n}\sum_{l=1}^{n} \tilde{\zeta}_h(x; x'_l)$;
    end for
    return $\{x_1, \cdots, x_N\}$;
end function
```

Note that although Algorithm 1 and 2 focus on the N samples generated in the final iteration, there is no reason to discard the intermediate samples for most practical purposes. That is, after j iterations, the algorithms will have generated $j \times N$ samples, and one can use all the $j \times N$ samples—as opposed to just N samples from the final iteration—for objectives (O1)-(O3) in the introduction, for example. Obviously, the empirical distributions of the samples from the earlier iterations can potentially be significantly different from the later iterations, but as one can see, for example, from Figure 9 as well as the consistency/convergence analysis in Section 3, the algorithm stabilizes and keeps producing samples (approximately) from the target distribution in the later iterations. Therefore, as the number of iterations j grows, the distribution of the overall $j \times N$ samples generated throughout the entire process will get closer to the target distribution.

3 Consistency and Convergence

In this section, we provide sufficient conditions for the convergence of the proposed algorithms. We make the following assumptions:

A1. f is $C_b^2(\mathcal{P})$;

A2. $\partial_i J_m f$ vanishes on the boundary $\partial\mathcal{P}$ of $\mathcal{P}$ for each $i = 1, \dots, m$.

The above conditions are imposed for the purpose of facilitating the convergence proof. We expect that Algorithm 1 and Algorithm 2 are consistent under much more general conditions. A2 simplifies the analysis since it allows $\tilde{\zeta}_h$ to produce consistent density estimation in terms of the derivative of the density in addition to the value of the density itself at the boundary; see (iii) of Lemma 2. If A2 does not hold, one can still prove the consistency by carefully dealing with the boundary of $\mathcal{P}$ separately as in Appendix B. In this section, we show that the empirical distribution of the points $\{x_1, \dots, x_N\}$ produced by Algorithm 2 "converges" to the target distribution ξ in $\mathcal{W}_1$ as the number of samples N and the iterations j grow. The precise statement will be given in Theorem 1. The consistency analysis of Algorithm 1 is more involved. We provide a discussion of Algorithm 1 in Appendix B.

Before starting the analysis, we construct a kernel that is consistent near the boundary. In simple words, we are defining these kernels with reflection along the boundaries. That is, if $\mathcal{P} = [0, 1]$, we fill in $[-1, 0]$ with the mirror images of the points on $[0, 1]$ (axis of reflection: $x = 0$) and also fill in $[1, 2]$ with the mirror images of the points on $[0, 1]$ (axis of reflection: $x = 1$) so that the data does not abruptly disappear outside of the boundary. This eliminates the boundary effect in the sense that the resulting kernel density estimation is consistent on the boundary; see (i) of Lemma 2. To be precise, consider a smooth and symmetric kernel ζ supported on $B(0; 1)$, such as biweight kernel, triweight kernel, and tricube kernel. We construct the kernel $\tilde{\zeta}_h$ from $\tilde{\zeta}_h^{(i)}(x; y)$, $i = 0, 1, \dots, m$, which are defined recursively as follows. Let $\tilde{\zeta}_h^{(0)}(x; y) \triangleq \zeta_h(x - y) \triangleq \frac{1}{h^m}\zeta\left(\frac{x-y}{h}\right)$ and define $\tilde{\zeta}_h^{(i)}$ for $i = 1, \dots, m$ as

$$\tilde{\zeta}_h^{(i+1)}(x; y) = \tilde{\zeta}_h^{(i)}(x; y) + \tilde{\zeta}_h^{(i)}\big(x; \operatorname{refl}_{\min}(y; i+1)\big) + \tilde{\zeta}_h^{(i)}\big(x; \operatorname{refl}_{\max}(y; i+1)\big),$$

where $x = (x^1, \ldots, x^m)^T$, $y = (y^1, \ldots, y^m)^T$, and

$$\mathrm{refl}_{\min}(y; i) \triangleq \begin{pmatrix} y^1 \\ \vdots \\ y^{i-1} \\ 2x^i_{\min} - y^i \\ y^{i+1} \\ \vdots \\ y^m \end{pmatrix} \quad \text{and} \quad \mathrm{refl}_{\max}(y; i) \triangleq \begin{pmatrix} y^1 \\ \vdots \\ y^{i-1} \\ -y^i + 2x^i_{\max} \\ y^{i+1} \\ \vdots \\ y^m \end{pmatrix}.$$

Finally, set

$$\tilde{\zeta}_h(x; y) \triangleq \tilde{\zeta}_h^{(m)}(x; y).$$

Note that for x, y such that $|x - y| > h$,

$$\tilde{\zeta}_h(x; y) = 0, \tag{3.1}$$

and for any $x \in \mathcal{P}$,

$$\int_{\mathcal{P}} \tilde{\zeta}_h(x; y) dy = 1. \tag{3.2}$$

For the purpose of the analysis, it is convenient to decompose each iteration of Algorithm 2 into four smaller conceptual pieces—smoothing step, smoothed sampling step, reweighting step, reweighted sampling step. At the beginning of the $(j+1)^{\text{th}}$ iteration, the algorithm starts with the empirical distribution $\hat{\xi}^{[j]} \triangleq \frac{1}{N} \sum_{i=1}^{N} \delta_{X_i^{[j]}}$ of samples $X_1^{[j]}, \ldots, X_N^{[j]}$ from the previous iteration. Note that for each design point $X_i^{[j]}$, we are using the superscript with square bracket for the iteration and subscript for the index of the design point within the iteration. Also, we are using the hat symbol in $\hat{\xi}^{[j]}$ to denote that it is a empirical measure of the samples generated from $\xi^{[j]}$. We will keep using these conventions throughout the paper. In the first step (smoothing step) of the iteration, the algorithm produces a probability density $\xi^{[j+1/2]}$ by smoothing and regularizing the empirical measure $\hat{\xi}^{[j]}$. In the second step (smoothed sampling step), the algorithm generates iid samples $X_1^{[j+1/2]}, \ldots, X_N^{[j+1/2]}$ from $\xi^{[j+1/2]}$ to obtain the empirical measure $\hat{\xi}^{[j+1/2]} \triangleq \frac{1}{N} \sum_{i=1}^{N} \delta_{X_i^{[j+1/2]}}$. In the third step (reweighting step), the algorithm adjusts the weight of each probability mass of the empirical distribution to get a new distribution $\xi^{[j+1]} \triangleq \frac{1}{N} \sum_{i=1}^{N} w_i \delta_{X_i^{[j+1/2]}}$, which has the same support as $\hat{\xi}^{[j+1/2]}$ but shifted (via redistribution of the weights w_i's) toward the target distribution. In the fourth step (reweighted sampling step), the algorithm generates samples $X_1^{[j+1]}, \ldots, X_N^{[j+1]}$ from $\xi^{[j+1]}$ and constructs a new empirical distribution $\hat{\xi}^{[j+1]}$. The first two steps correspond to the perturbation step in Algorithm 2 and the last two steps correspond to the resampling step in Algorithm 2. Schematically, the process can be summarized as follows:

$$\begin{array}{rccccccccc}
0^{\text{th}} \text{ iteration:} & & & & & \hat{\xi}^{[\ -1/2]} & \xrightarrow{\text{reweighting}} & \xi^{[0]} & \xrightarrow{\text{sampling}} & \hat{\xi}^{[0]} \\
1^{\text{st}} \text{ iteration:} & \hat{\xi}^{[0]} & \xrightarrow{\text{smoothing}} & \xi^{[0+1/2]} & \xrightarrow{\text{sampling}} & \hat{\xi}^{[0+1/2]} & \xrightarrow{\text{reweighting}} & \xi^{[1]} & \xrightarrow{\text{sampling}} & \hat{\xi}^{[1]} \\
2^{\text{nd}} \text{ iteration:} & \hat{\xi}^{[1]} & \xrightarrow{\text{smoothing}} & \xi^{[1+1/2]} & \xrightarrow{\text{sampling}} & \hat{\xi}^{[1+1/2]} & \xrightarrow{\text{reweighting}} & \xi^{[2]} & \xrightarrow{\text{sampling}} & \hat{\xi}^{[2]} \\
\vdots & & & & & & & & &
\end{array}$$

where $\hat{\xi}^{[\ -1/2]}$ is the empirical distribution of an arbitrary initial samples. A typical choice would be iid uniform samples from $\mathcal{P}$. The precise description of the four steps is as follows: to generate samples from the target measure ξ on the input space $\mathcal{P}$, (or equivalently, η on the manifold $\mathcal{M}$),

- At iteration 0, we skip the smoothing and smoothed sampling step, and start directly with an arbitrary initial samples $X_1^{[\ -1/2]}, \dots, X_N^{[\ -1/2]}$. (Then we proceed to the reweighting step and the reweighted sampling step.)
- At iteration $j+1$, we start with an empirical distribution $\hat{\xi}^{[j]}$ of the samples $X_1^{[j]}, \dots, X_N^{[j]}$ from the previous iteration

  $$\hat{\xi}^{[j]} \triangleq \frac{1}{N} \sum_{i=1}^{N} \delta_{X_i^{[j]}}.$$

  Now, for suitably chosen parameters h and q (whose choice will be discussed later in this section and Section 4),

  Step 1) Smooth out the empirical distribution $\hat{\xi}^{[j]}$ with ζ_h to get $\tilde{\xi}^{[j+1/2]}$

  $$\frac{d\tilde{\xi}^{[j+1/2]}}{d\lambda^m}(x) \triangleq \frac{1}{N} \sum_{i=1}^{N} \tilde{\zeta}_h(x; X_i^{[j]}).$$

  Set $\xi^{[j+1/2]}$ as a mixture of the uniform distribution (on $\mathcal{P}$) and $\tilde{\xi}^{[j+1/2]}$ with probability q and $1-q$, respectively:

  $$\frac{d\xi^{[j+1/2]}}{d\lambda^m}(x) \triangleq q/\lambda^m(\mathcal{P}) + (1-q)\frac{1}{N} \sum_{i=1}^{N} \tilde{\zeta}_h(x; X_i^{[j]})$$

  where $\lambda^m(\mathcal{P})$ denotes the m dimensional volume of $\mathcal{P}$.

  Step 2) Generate iid samples $X_1^{[j+1/2]}, \dots, X_N^{[j+1/2]}$ from $\xi^{[j+1/2]}$, and set $\hat{\xi}^{[j+1/2]}$ to be the empirical distribution of the generated samples:

  $$\hat{\xi}^{[j+1/2]} \triangleq \frac{1}{N} \sum_{i=1}^{N} \delta_{X_i^{[j+1/2]}}.$$

Step 3) Evaluate f and $J_m f$ at $X_i^{[j]}$-s and re-distribute the weights as follows:

$$\xi^{[j+1]} \triangleq \sum_{i=1}^{N} \frac{(\mu \circ f \cdot J_m f \cdot \iota)(X_i^{[j+1/2]})}{\sum_{l=1}^{N}(\mu \circ f \cdot J_m f \cdot \iota)(X_l^{[j+1/2]})} \delta_{X_i^{[j+1/2]}}, \tag{3.3}$$

where $\iota \triangleq d\lambda^m / d\xi^{[j+1/2]}$ is the reciprocal of the density of $\xi^{[j+1/2]}$ w.r.t. the Lebesgue measure. Recall that μ is the Radon-Nikodym derivative of the target distribution η.

Step 4) Generate iid samples $X_1^{[j+1]}, \ldots, X_N^{[j+1]}$ from $\xi^{[j+1]}$ to get an empirical distribution

$$\hat{\xi}^{[j+1]} \triangleq \frac{1}{N} \sum_{i=1}^{N} \delta_{X_i^{[j+1]}}$$

- Repeat step 1)-4) to get $\hat{\xi}^{[j+2]}, \hat{\xi}^{[j+3]}, \ldots$

Remark. *Note that sampling from $\xi^{[j+1/2]}$ in Step 2 can be implemented by sampling from uniform distribution in $\mathcal{P}$ with probability q and perturbing $X_i^{[j]}$ according to the kernel $\tilde{\zeta}_h$ with probability $(1-q)/N$ for each $i = 1, \ldots, N$.*

The main result of this section is that the above algorithm is consistent. Let

$$\alpha(N) \triangleq \begin{cases} N^{-1/2} & \text{if } m = 1 \\ N^{-1/2} \log(N+1) & \text{if } m = 2 \\ N^{-1/m} & \text{otherwise} \end{cases}$$

and, for any function $g : A \to \mathbb{R}^d$ on a domain A, let $\|g\|_\infty \triangleq \sup_{x \in A} |g(x)|$ where $|\cdot|$ is the Euclidean norm, and let $\|g\|_{\text{Lip}} \triangleq \sup_{x,y \in A,\, x \neq y} \frac{|g(x)-g(y)|}{|x-y|}$. Let $D \triangleq \operatorname{diam}(\mathcal{P})$, $V \triangleq \lambda^m(\mathcal{P})$, and c denote the constant from Proposition 2 that depends only on $\|d\xi/d\lambda^m\|_\infty$, $\|d\xi/d\lambda^m\|_{\text{Lip}}$, $\|\nabla(d\xi/d\lambda^m)\|_{\text{Lip}}$, q, and m.

Theorem 1. *Suppose that h is chosen in such a way that*

$$\phi(h,N) \triangleq c\alpha(N)\big(D^2V^2(\|\zeta_h\|_{\text{Lip}} + \|\nabla\zeta_h\|_{\text{Lip}}) + DV\|\zeta_h\|_{\text{Lip}}\big) < 1. \tag{3.4}$$

Algorithm 2 is consistent in the sense that the Kantorovich-Rubinstein distance between the output $\hat{\xi}^{[j]}$ and the target measure ξ converges to 0 in L_1 as $N \to \infty$ and $j \to \infty$. More specifically,

$$\mathbf{E}\,\mathcal{W}_1(\hat{\xi}^{[j]}, \xi) \leq \frac{c\alpha(N)\big((D^2V^2 + DV)h + (2D + D^2V)\big)}{1 - \phi(h,N)} + \phi(h,N)^j \mathbf{E}\,\mathcal{W}_1(\hat{\xi}^{[0]}, \xi). \tag{3.5}$$

We defer the proof of Theorem 1 to Appendix A and conclude this section with a few remarks regarding the implications of the theorem.

Note first that the second term on the RHS of (3.5) decays at a geometric rate as j increases, whereas the first term does not depend on j. Also, the ratio between the first term and the geometric factor of the second term is

$$\frac{\big((D^2V^2 + DV)h + (2D + D^2V)\big)/\big(D^2V^2(\|\zeta_h\|_{\mathrm{Lip}} + \|\nabla\zeta_h\|_{\mathrm{Lip}}) + DV\|\zeta_h\|_{\mathrm{Lip}}\big)}{1 - c\alpha(N)\big(D^2V^2(\|\zeta_h\|_{\mathrm{Lip}} + \|\nabla\zeta_h\|_{\mathrm{Lip}}) + DV\|\zeta_h\|_{\mathrm{Lip}}\big)}.$$

Since $\|\zeta_h\|_{\mathrm{Lip}}$ and $\|\nabla\zeta_h\|_{\mathrm{Lip}}$ are typically $c_1 \cdot h^{-m-1}$ and $c_2 \cdot h^{-m-2}$, respectively, for some constants c_1 and c_2 that are greater than 1, the ratio is bounded by

$$h^{m+2}\frac{\big((1 + D^{-1}V^{-1})h + (2D^{-1}V^{-2} + V^{-1})\big)/\big(1 + (1 + D^{-1}V^{-1})h\big)}{1 - c\alpha(N)\big(D^2V^2(\|\zeta_h\|_{\mathrm{Lip}} + \|\nabla\zeta_h\|_{\mathrm{Lip}}) + DV\|\zeta_h\|_{\mathrm{Lip}}\big)}.$$

In case D and V are large (e.g., in our running example (Example 1.1), $D = 100\sqrt{2}$ and $V = 10,000$), and N and h are chosen so that the geometric factor $\phi(h, N)$ in (3.4) is sufficiently away from 1, the ratio is of order h^{m+2}. On the other hand, note that $\mathbf{E}\,\mathcal{W}_1(\hat{\xi}^{[0]}, \xi)$ is typically of order D. Therefore, the second term in (3.5) will dominate the error for small j. But as j increases, the first term will dominate the error. That is, as the number j of iterations increases, $\hat{\xi}^{[j]}$ will approach ξ geometrically fast at the beginning, but then the convergence will eventually slow down and $\hat{\xi}^{[j]}$ will linger around ξ. In view of this, one should choose the number of iterations j in such a way that the two terms in (3.5) are of similar size. That is, for a given N, j should be chosen roughly at around

$$\frac{\log \mathbf{E}\,\mathcal{W}_1(\hat{\xi}^{[0]}, \xi) - \log\Big(c\alpha(N)\big((D^2V^2 + DV)h + (2D + D^2V)\big)\Big)}{-\log\Big(c\alpha(N)\big(D^2V^2(\|\zeta_h\|_{\mathrm{Lip}} + \|\nabla\zeta_h\|_{\mathrm{Lip}}) + DV\|\zeta_h\|_{\mathrm{Lip}}\big)\Big)}. \tag{3.6}$$

If j is much smaller than (3.6), the second term will dominate the final error even though it can be reduced geometrically and hence losing opportunities to reduce the total error efficiently; on the other hand, if j is much larger than (3.6), the first term will dominate the error and hence one will waste extra efforts without much gain in terms of the error. Note also that if the starting configuration $\hat{\xi}^{[0]}$ is reasonably close to ξ to begin with, Algorithm 2 will require very small number of iterations before it stabilizes. This is consistent with our numerical experiences reported in Section 4.

There is a tradeoff between choosing h small and large. Note that the condition (3.4) requires that h should not decrease to 0 too fast compared to the rate at which the size N of samples grows. On the other hand, in case D and V are large, the first term in (3.5), whch does not decrease at a geometric rate w.r.t. the number of iterations, will be dominated by $c\alpha(N)D^2V^2h$ for the practical range of values of N, and hence, small h is desired. That is, larger h allows to make sure that the algorithm is stable w.r.t. the iterations and for smaller number of samples N; smaller h allows one to reduce the final error of the algorithm after a sufficiently large number of iterations.

While our convergence bound is explicit, such explicitness comes at the expense of tight asymptotics. Recall that $\|\zeta_h\|_{\mathrm{Lip}}$ and $\|\nabla\zeta_h\|_{\mathrm{Lip}}$ are typically of order h^{-m-1} and h^{-m-2}, respectively, and hence, h should not decrease to 0 at a faster rate than $N^{-\frac{1}{m(m+2)}}$ to satisfy condition (3.4). This is a very slow rate even for moderately large m's, and we expect that the algorithm would be stable for h's that decreases faster than such a rate guaranteed by Theorem 1. The practical choice of h and q will be further discussed in Section 4.

Finally, we point out that the decay rate of $\alpha(N)$ slows down as the dimensionality m grows. This is an inherent difficulty that no space-filling scheme can avoid. The specific form of the dependency on the dimensionality reflects the fact that even when one can generate the iid samples from the exact target distribution, the Kantorovich-Rubinstein distance between the empirical distribution and the target distribution will decrease at the same rate.

4 Examples

In this section, we briefly discuss the choice of algorithmic parameters h, q, and b, and examine the numerical behavior of the algorithm with a few examples. Due to the conservative nature of our convergence analysis in Theorem 1 and Lemma 3, we do not provide definitive rules for the choice of the above parameters. Instead, we provide heuristic discussions and rough guidelines from our numerical experience here. More thorough investigation will be pursued in subsequent studies. As pointed out in Section 3, the choice of h is critical for the stability and the performance of the algorithm. Although our sufficient condition in Theorem 1 is conservative, our numerical experience confirms that a liberal choice of h can indeed lead to unstable behavior of the algorithm. That is, if h is chosen too small compared to N, the algorithm may diverge from the target distribution. An indication of such divergence is clustering of the points, which can easily be detected with various methods. In view of this, we suggest starting with a conservative choice of h and gradually reducing the size of h as the algorithm stabilizes. When the clustering behavior is detected, the experimenter can either increase N or increase h so that the algorithm does not exhibit clustering behavior. Turning to q, note that q being away from 0 prevents $\xi^{[j+1/2]}$ from being much smaller than ξ so that the ratio doesn't blow up. On the other hand, if q is close to 1, our algorithm is not assigning enough resource (samples) in learning the geometry of the manifold. Such a trade-off can be noticed in the upper bounds in (ii) and (iv) of Lemma 2 where both q and $1-q$ appear in the denominator. In view of this, we suggest choosing q inside of the interior of $(0,1)$ sufficiently away from the boundary, say, between $1/10$ and $1/2$. The choice of b does not seem to make much difference in terms of the performance of the algorithm as far as b is chosen sufficiently large.

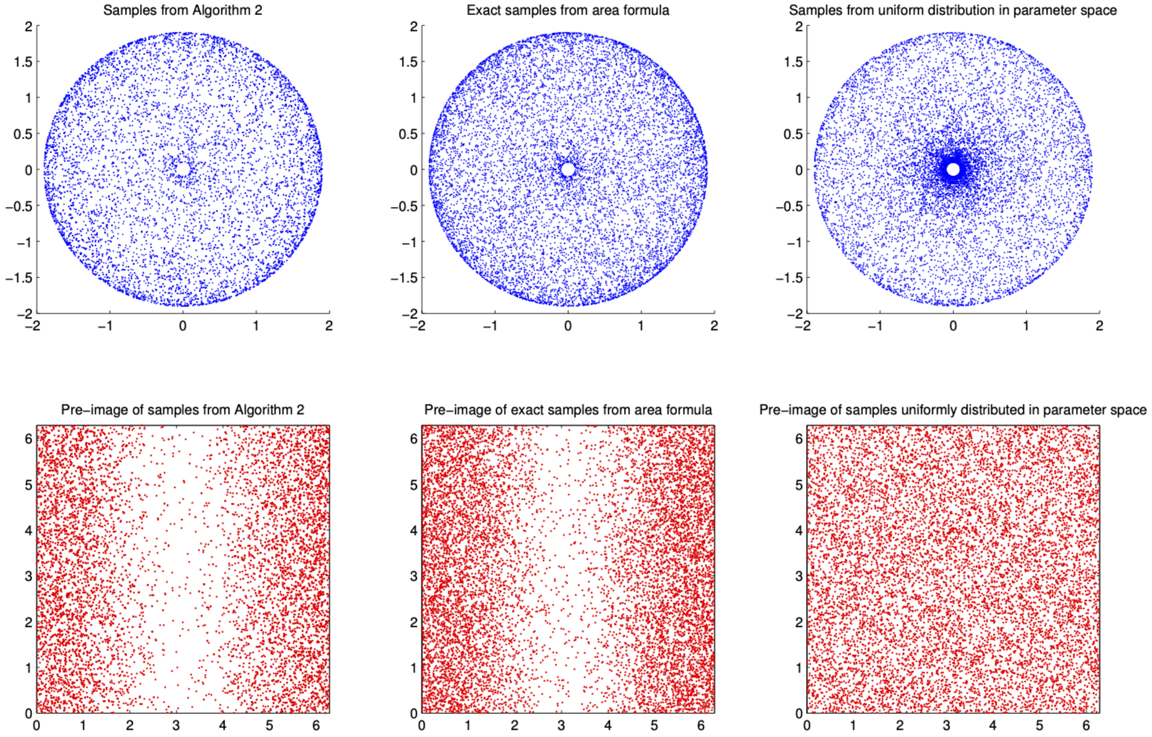

Figure 3: (Uniform Samples From Torus) Comparison of the 10,000 samples from Algorithm 2, exact uniform distribution on the manifold, and uniform distribution in the input space.

Example 1. (Uniform Samples from Torus) Diaconis et al. (2013) illustrate how to sample from a torus using the area formula (2.2). Consider a torus

$$\mathcal{M} = \{((R + r\cos\theta)\cos\psi, (R + r\cos\theta)\sin\psi, r\sin\theta) : 0 \le \theta, \psi < 2\pi\}$$

where $0 < r < R$. The major radius R is the distance from the center of the tube to the center of the torus, and the minor radius r is the radius of the tube. One way to parametrize $\mathcal{M}$ and its 2-dimensional Jacobian $J_2 f$ are

$$f(\theta, \psi) = ((R + r\cos\theta)\cos\psi, (R + r\cos\theta)\sin\psi, r\sin\theta), \qquad J_2 f(\theta, \psi) = r(R + r\cos\theta).$$

In view of (2.2), one can generate exactly uniform samples on $\mathcal{M}$ w.r.t. Hausdorff measure by generating samples on $[0, 2\pi] \times [0, 2\pi]$ from the density $g(\theta, \psi) \propto R + r\cos\theta$. For example, one

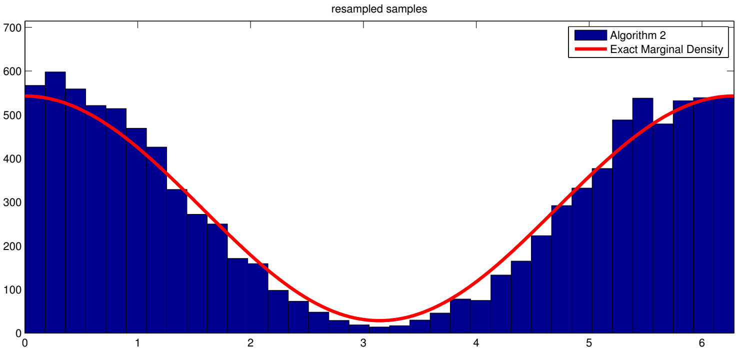

Figure 4: (Uniform Samples From Torus) Histogram from 10,000 samples of θ's generated by Algorithm 2. Red line shows the exact target marginal density computed from the area formula.

can generate ψ from the uniform distribution on $[0, 2\pi]$, and (independently) generate θ from the density $\frac{1}{2\pi R}(R + r\cos\theta)$, via acceptance-rejection or inversion. We compare the three different ways of covering $\mathcal{M}$ for the purpose of illustration of the consistency of our algorithm. Figure 3 compares the samples on $\mathcal{M}$ produced for $R = 1$ and $r = 0.9$. The upper plot shows the samples projected on x-y plane, and the lower plot shows the pre-image of the samples in the input space. The plots on the left show the samples generated from Algorithm 2 after the resampling step in the second iteration with $q = 0.1$ and $h = 0.5$. The plots in the middle were produced with the 10,000 samples generated by the area formula (as described above), and the right plots show 10,000 samples generated by uniformly sampling in the input space, i.e., $\theta \sim U[0, 2\pi]$ and $\psi \sim U[0, 2\pi]$. Observe that the left and middle plots are similar to each other while in the upper right plot the center of the torus is much more densely populated compared to the outer part of the torus. This illustrates that the samples generated uniformly from the input space A is far from uniform on the manifold $\mathcal{M}$, and Algorithm 2 produces samples from the target distribution. Figure 4 compares the histogram of θ sampled by Algorithm 2, and the exact marginal of the target density $g(\theta, \psi) = \frac{1}{4\pi^2 R}(R + r\cos\theta)$.

Example 2. (Non-uniform Density on Torus) In this example, we consider the same manifold $\mathcal{M}$ as in Example 1, but we illustrate how Algorithm 2 performs for a non-uniform density on $\mathcal{M}$. Suppose that we are particularly interested in studying $\mathcal{M}$ in the proximity of a given point. For example, suppose that we are interested in $(0, 1, 0)$, and hence, we want to use more computational resource for the closer parts of the manifold to the point, and less resource for the farther parts of the manifold. For this purpose, we choose a density proportional to the reciprocal of the squared distance from $(0, 1, 0)$. More specifically, we want to sample from the distribution $P(d\mathbf{x}) = r(\mathbf{x})\mathcal{H}^2(d\mathbf{x})$ where $r(x, y, z) \propto 1/(x^2 + (y - 1)^2 + z^2)$. Again, for this simple example, one can generate exact samples from P directly from the area formula via acceptance-rejection with the proposal density proportional to

$$r(f(\theta,\psi))g(\theta,\psi) \propto \frac{R + r\cos\theta}{((R + r\cos\theta)\cos\psi)^2 + ((R + r\cos\theta)\sin\psi - 1)^2 + (r\sin\theta)^2}.$$

The samples produced by Algorithm 2 (after the third resampling step with $q = 0.1$ and $h = 0.5$), the exact samples generated by the area formula, and the samples generated uniformly in the input space are compared in Figure 5 and 6. Once can again it should be noted that Algorithm 2 generates the correct distribution.

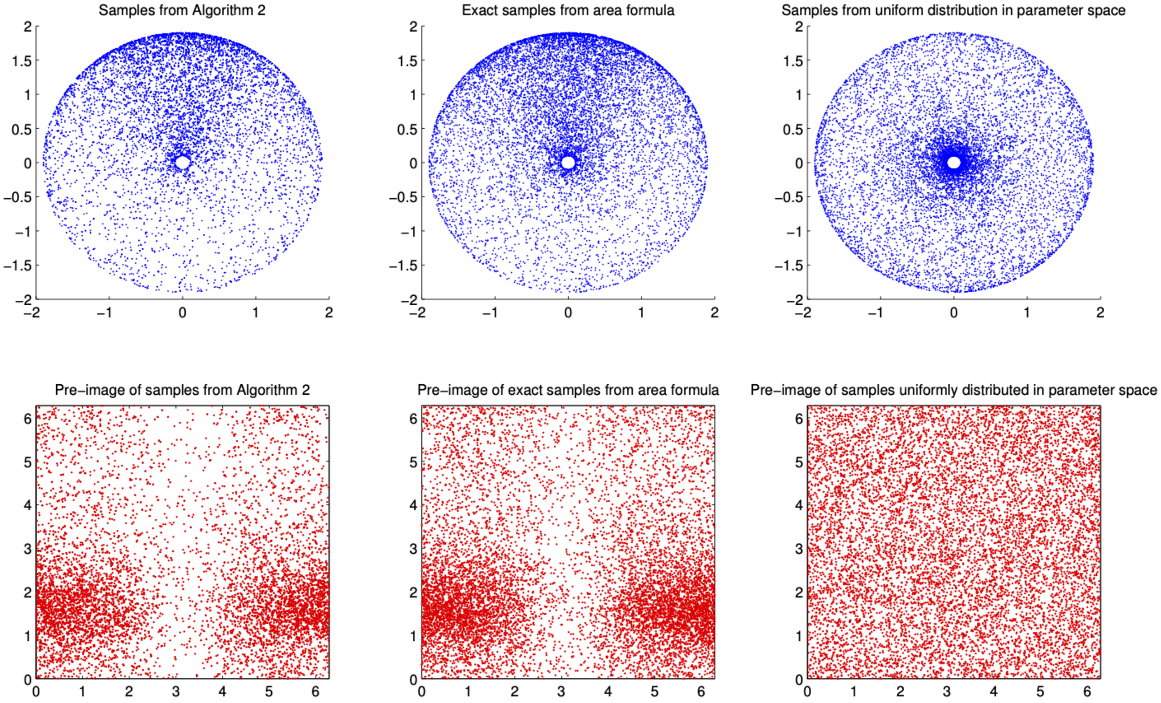

Figure 5: (Non-Uniform Density on Torus) Comparison of the 10,000 samples from Algorithm 2, area formula, and uniform distribution in the input space.

Next, we examine a more interesting case, where the model changes its behavior significantly on a small part of the input space while it remains relatively constant over the majority of the input space. That is, most of the input space is mapped to a small fraction of the manifold and the rest—the majority—of the manifold comes from a small fraction of the input space. The next

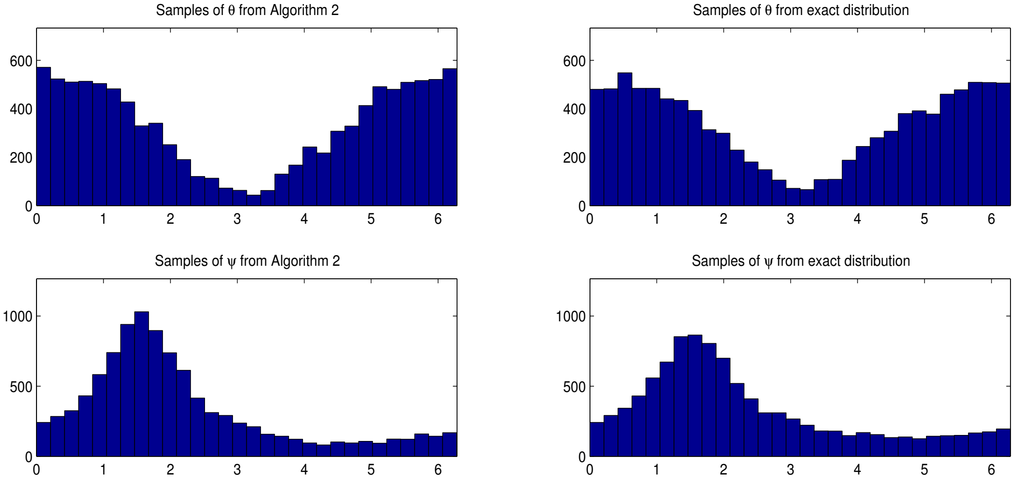

Figure 6: (Non-Uniform Density on Torus) Histograms from 10,000 samples generated by Algorithm 2, and the exact area formula.

example illustrates how our algorithm discovers such a small region of the input space.

Example 3. (Exponential Model) Here we consider a manifold

$$\mathcal{M} = \{(e^{-\theta t_1} + e^{-\psi t_1}, e^{-\theta t_2} + e^{-\psi t_2}, e^{-\theta t_3} + e^{-\psi t_3}) : \theta, \psi \in [0, 100]\}$$

where $0 < t_1 < t_2 < t_3$. The first derivative

$$Df(\theta, \psi) = \begin{pmatrix} -t_1 \exp(-\theta t_1) & -t_1 \exp(-\psi t_1) \\ -t_2 \exp(-\theta t_2) & -t_2 \exp(-\psi t_2) \\ -t_3 \exp(-\theta t_3) & -t_3 \exp(-\psi t_3) \end{pmatrix}, \tag{4.1}$$

and the 2-dimensional Jacobian

$$J_2 f(\theta, \psi) = \sqrt{\alpha_{12} + \alpha_{13} + \alpha_{23}} \tag{4.2}$$

where

$$\alpha_{ij} = t_i^2 t_j^2 \exp\{-2(\theta t_j + \psi t_i)\}\{\exp(\theta - \psi)(t_j - t_i) - 1\}^2. \tag{4.3}$$

Figure 7 shows the result for $t_1 = 1$, $t_2 = 2$, $t_3 = 4$. The left plot was produced by Algorithm 2 with $q = 0.1$ and $h = 1$, (the upper plot shows the samples projected on a plane perpendicular to the vector $(0.5, -1, 0.5)$, and the lower plot shows the pre-image of the samples in the input space) with 2,000 particles after 10 iterations (hence, total 20,000 evaluations of $J_m f$ throughout the whole process); the plots in the middle show the 2,000 samples generated by the area formula; the right plots show 60,000 samples generated by uniformly sampling in the input space, i.e., $\theta, \psi \sim \text{Unif}\big([0,100] \times [0,100]\big)$. Note that the evaluation of f requires 3 scalar-valued function evaluations, and the evaluation of $J_2 f$ requires 6 scalar-valued function evaluations, and hence, the amount of computation required Algorithm 2 and uniform sampling in the input space to generate the Figure 7 are comparable. The samples from Algorithm 2 covers the target manifold well. In contrast, the samples generated uniformly in the input space are concentrated in a small part of the manifold and fail to discover vast majority of it. The contrast becomes even more stark when we consider the fact that for most practical purposes we will keep all the 20,000 particles generated throughout all 10 iterations, which will provide even better coverage of the manifold with Algorithm 2.

We started the Algorithm 2 with initial samples distributed uniformly in the input space, and the final samples were obtained after the resampling step in the 10th iteration. The progression of the algorithm is illustrated in Figure 9.

Example 4. (ODE Models in Systems Biology) The dynamics of the enzymatic regulatory systems are often modeled with a set of ordinary differential equations. One of the most popular form of such differential equations is Michaelis-Menten kinetics (Michaelis et al., 2011). Consider the following Michaelis-Menten kinetics between three different kinds of enzymes A, B, C, and the input I:

$$\begin{aligned}
\frac{dA}{dt} &= k_{IA} I \frac{(1-A)}{(1-A)+K_{IA}} - F_A k'_{F_A A} \frac{A}{A + K'_{F_A A}} \\
\frac{dB}{dt} &= C k_{CB} \frac{(1-B)}{(1-B)+K_{CB}} - F_B k'_{F_B B} \frac{B}{B + K'_{F_B B}} \\
\frac{dC}{dt} &= A k_{AC} \frac{(1-C)}{(1-C)+K_{AC}} - B k'_{BC} \frac{C}{C + K'_{BC}}.
\end{aligned} \tag{4.4}$$

Assume that the exact values of k_{IA} and k_{CB} are unknown. One way to proceed to study the model is to sample k_{IA} and k_{CB} randomly from a plausible range, say $\mathcal{P} \triangleq [d_1, u_1] \times [d_2, u_2]$, and see if the model can exhibit the desired behavior of the enzymatic system. A typical approach in systems biology is to sample a number of input parameters uniformly from R and observe what kind of model behaviors are exhibited at the selected design points (Ma et al., 2009). However, the change of dynamics w.r.t. the change of the values of k_{IA} and k_{CB} might be highly non-linear so that the observation based on insufficient number of uniform samples can be misleading. Suppose that we are interested in the adaptive behavior of the model. Adaptation refers to the ability of the system to respond (i.e., change the output level) to an input stimulus (i.e., change in input level), and then return to its original output level even when the change in the input level persists. The adaptive behavior can be summarized as the *sensitivity* and the *precision* of the system. In the context of

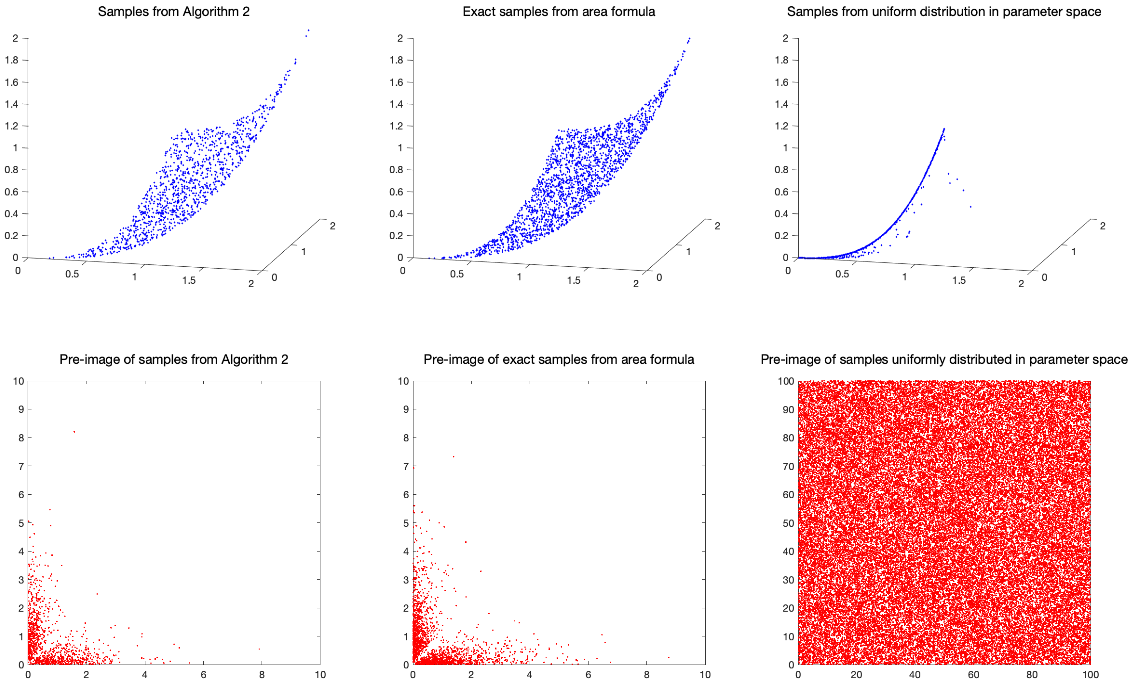

Figure 7: (Exponential Model) Comparison of the 2,000 samples from Algorithm 2 after 10 iterations (hence total 20,000 evaluations of f and $J_2 f$), 2,000 samples from area formula, and 60,000 samples from the uniform distribution in input space.

our example (4.4), the sensitivity is defined as the ratio $\left|\frac{(C_{\text{peak}}-C_0)/C_0}{(I_1-I_0)/I_0}\right|$ between the size of the response of the output C and the size of the stimulus (i.e., the change in the input I), where I_0 and C_0 are the initial input and output levels respectively, I_1 is the new input level, and C_{peak} is the maximum of the output level after the intput level changes from I_0 to I_1; the precision is defined as the ratio $\left|\frac{(I_1-I_0)/I_0}{(C_1-C_0)/C_0}\right|$ between the (long term) change in the input and output levels, where C_1 is the final output level of the system by the time the system stabilizes after the initial change due to the stimulus. These output measures quantify how well the system detects the change in the input level, and how robust is the system to such change, respectively. For more thorough description of adaptation, sensitivity, and precision, see Ma et al. (2009). For our purpose, just note that ODE in (4.4) defines a mapping from the input space $\mathcal{P}$ to the output space $\mathbb{R}^2$, each coordinate of which represents the sensitivity and the precision, respectively. Figure 10 compares the observations based

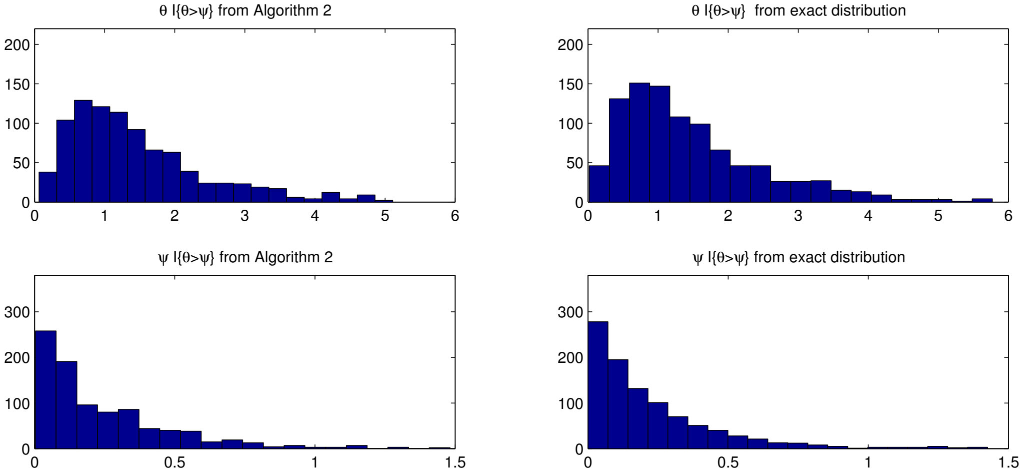

Figure 8: (Exponential Model) Histograms from 2,000 samples generated by Algorithm 2 and the exact area formula.

on uniform sampling on the input space and observations based on uniform samples on the output space. More specifically, the right plot shows the observation based on sampling $(\log_{10} k_{IA} + 1)/2$ and $(\log_{10} k_{CB} + 1)/2$ uniformly from $R = [0.35, 0.88] \times [0, 1]$, and the other two plots show the observations based on sampling uniformly from the output space via Algorithm 1 with $q = 0.1$, $k = 5$, $h = 0.03$, $b = \infty$, and $N = 1000$. We used Algorithm 1 instead of Algorithm 2 in this example, since the exact derivative of the mapping is not readily available. The coefficients of the ODE (other than k_{IA} and k_{CB}) were chosen as follows:

$$\begin{pmatrix} k'_F AA \\ k'_F BB \\ k_A C \\ k'_B C \\ K_I A \\ K'_F AA \\ K_C B \\ K'_F BB \\ K_A C \\ K'_B C \end{pmatrix} = \begin{pmatrix} 7.0437 \\ 0.1364 \\ 3.0061 \\ 0.8395 \\ 0.0183 \\ 0.0016 \\ 0.0122 \\ 0.0032 \\ 0.0044 \\ 0.0742 \end{pmatrix}.$$

One can see that by uniformly sampling on the input space one may end up wasting lots of design points to explore the lower left part of the model output space while almost missing the protruding

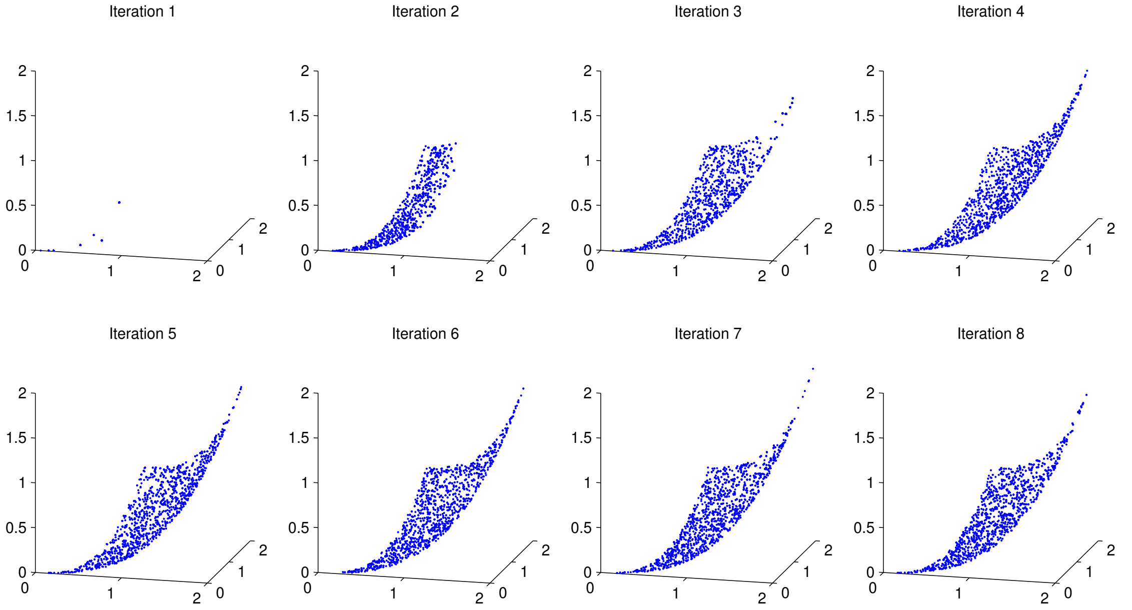

Figure 9: (Exponential Model) The progression of Algorithm 2.

region on the lower right part of the model output space. On the other hand, our algorithm distributes the design points intelligently so that the nearly missed lower right part of the model output space is clearly identified with less total number (4000 times) of ODE simulations compared to the naïve design points uniform in the input space (5000 times).

Acknowledgement

The first author gratefully acknowledges the support from National Science Foundation (CMMI-2146530). The second author is grateful for the support by AFOSR under Grant FA9550-22-1-0244 and National Science Foundation under Grant DMS2053489. The third author is grateful for the support by National Science Foundation under Grant CCF1552784.

A numerical investigation (without convergence analysis) of the preliminary version of Algorithm 1 in this paper has been presented in a conference proceeding Rhee et al. (2014). The most significant difference between the algorithm proposed in Rhee et al. (2014) and the ones in the current paper is in the perturbation step. While the perturbation step was performed by simulating diffusions in the algorithm proposed in Rhee et al. (2014), Algorithm 1 in the current paper smoothes the current empirical distribution with a Kernel density estimator and mixes the smoothed

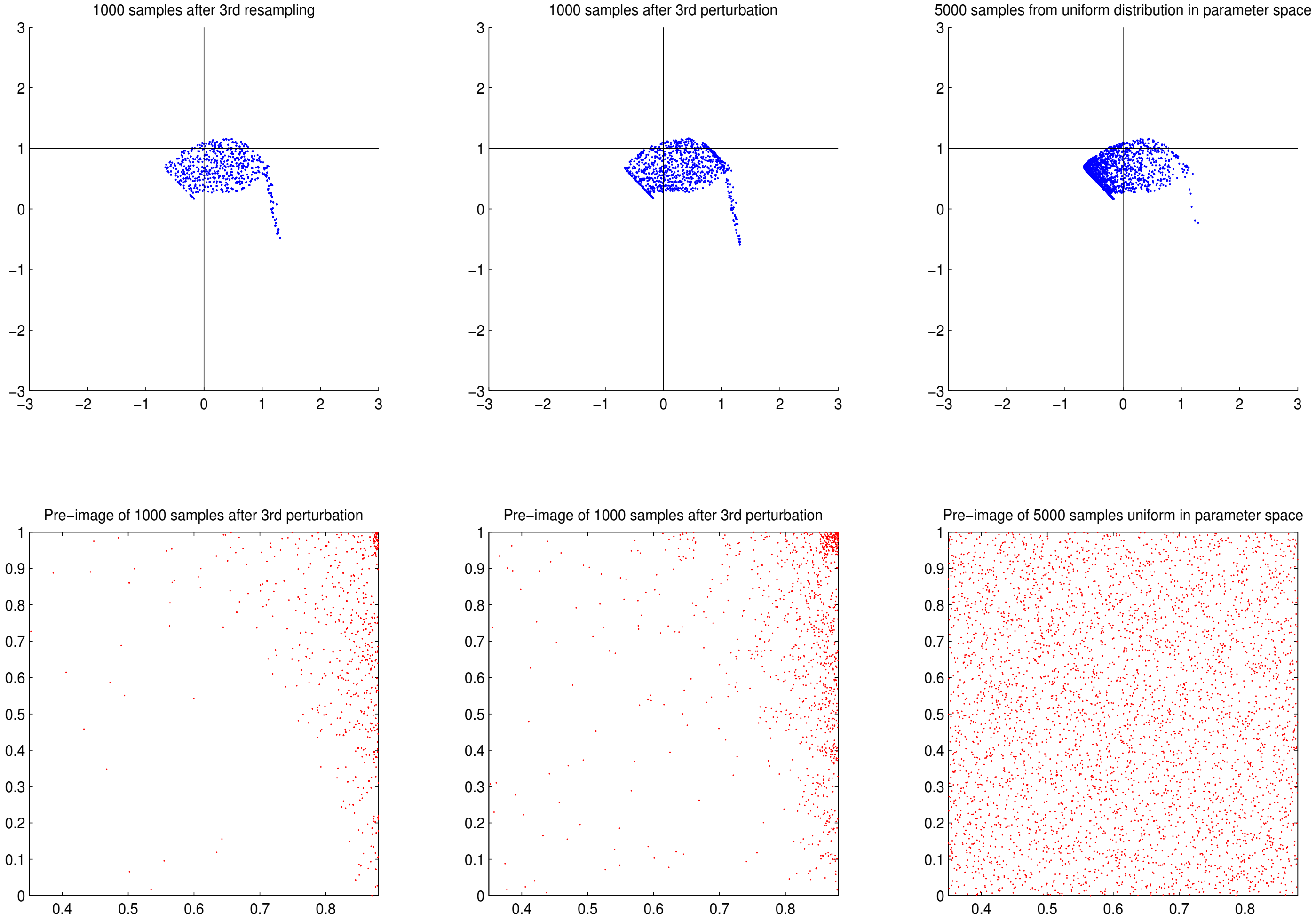

Figure 10: The right plot shows the observation based on the samples uniformly distributed on $\mathcal{P} = [0.35, 0.88] \times [0, 1]$, and the other two plots show the observations based on Algorithm 1.

distribution with a uniform distribution. Such a change in the perturbation step reduces the computational burden and makes it possible to implement Algorithm 2 since the density of the perturbed samples become analytically tractable. Moreover, we prove the consistency and convergence of the proposed algorithm in this paper.

A Appendix: Proof of Theorem 1

This section provides the proof of Theorem 1. For the notational convenience, we adopt the (common) convention that for a function $h : A \to \mathbb{R}$ and a measure μ on A, we denote the integral of h w.r.t. μ with $\mu(h)$ or μh:

$$\mu(h) \triangleq \mu h \triangleq \int_A h(y)\mu(dy).$$

For example, $\Psi_G(\xi)f = \int f(x)\Psi_G(\xi)(dx) = \int f(x)G(x)\xi(dx)/\int G(x)\xi(dx) = \xi(fG)/\xi(G)$ with this notation. The proof of Theorem 1 hinges critically on the following proposition.

Proposition 2. *There exists a constant c depending only on $\|d\xi/d\lambda^m\|_\infty$, $\|d\xi/d\lambda^m\|_{\mathrm{Lip}}$, $\|\nabla(d\xi/d\lambda^m)\|_{\mathrm{Lip}}$, q, and m such that*

$$\begin{aligned}\mathbf{E}\,\mathcal{W}_1(\xi^{[j+1]},\xi) \le c\alpha(N)\big(D^2V^2(\|\zeta_h\|_{\mathrm{Lip}} + \|\nabla\zeta_h\|_{\mathrm{Lip}}) + DV\|\zeta_h\|_{\mathrm{Lip}}\big)\mathbf{E}\,\mathcal{W}_1(\xi^{[j]},\xi)\\ + c\alpha(N)\big((D^2V^2 + DV)h + (D + D^2V)\big)\end{aligned}$$

where $D = \operatorname{diam}(\mathcal{P})$ and $V = \lambda^m(\mathcal{P})$.

Before proving Proposition 2, we first establish the following key lemmas. Note that $\xi^{[j+1]} = \Psi_G(\hat{\xi}^{[j+1/2]})$ and $\xi = \Psi_G(\xi^{[j+1/2]})$, where $G = (d\xi/d\lambda^m)/(d\xi^{[j+1/2]}/d\lambda^m)$. In view of this, it is important to understand how smooth $\Psi_G(\nu)$ is w.r.t. ν in terms of $\mathcal{W}_1$ distance. Lemma 1 provides a useful bound in this regard, and it turns out that the bound involves quantities $\|d\xi/d\xi^{[j+1/2]}\|_\infty$ and $\|d\xi/d\xi^{[j+1/2]}\|_{\mathrm{Lip}}$. Lemma 2 provides estimates for these quantities.

Lemma 1.

$$\mathcal{W}_1(\xi^{[j+1]},\xi) \le \big(\|d\xi/d\xi^{[j+1/2]}\|_\infty + 2\operatorname{diam}(\mathcal{P})\cdot\|d\xi/d\xi^{[j+1/2]}\|_{\mathrm{Lip}}\big)\,\mathcal{W}_1(\xi^{[j+1/2]},\hat{\xi}^{[j+1/2]}). \tag{A.1}$$

Proof. Set $\hat{\nu} \triangleq \hat{\xi}^{[j+1/2]}$, $\nu \triangleq \xi^{[j+1/2]}$, and $G \triangleq (d\xi/d\lambda^m)/(d\nu/d\lambda^m)$. For the notational simplicity, we slighly abuse the notation and denote the densities of ν and ξ w.r.t. the Lebesque measure with ν and ξ as well; for example, we will write $G(x) = \xi(x)/\nu(x)$. Note that with these notations,

$\Psi_G(\hat{\nu}) = \xi^{[j+1]}$, $\Psi_G(\nu) = \xi$, and $\nu(G) = 1$. Also, for any function f,

$$\begin{aligned}\Psi_G(\hat{\nu})f - \Psi_G(\nu)f &= \frac{\hat{\nu}(Gf)}{\hat{\nu}(G)} - \frac{\nu(Gf)}{\nu(G)} \\ &= \frac{(\nu(G) - \hat{\nu}(G))\hat{\nu}(Gf) + \hat{\nu}(G)(\hat{\nu}(Gf) - \nu(Gf))}{\hat{\nu}(G)\nu(G)} \\ &= (\nu - \hat{\nu})(G)\Psi_G(\hat{\nu})f + (\hat{\nu} - \nu)(Gf)\end{aligned}$$

Therefore,

$$\begin{aligned}\mathcal{W}_1(\Psi_G(\hat{\nu}), \Psi_G(\nu)) &= \sup\{\Psi_G(\hat{\nu})f - \Psi_G(\nu)f : \|f\|_{\text{Lip}} \leq 1, f(x_0) = 0\} \\ &= \sup\{(\hat{\nu} - \nu)(Gf) + (\nu - \hat{\nu})(G)\Psi_G(\hat{\nu})f : \|f\|_{\text{Lip}} \leq 1, f(x^0) = 0\} \\ &\leq \sup\{\hat{\nu}(Gf) - \nu(Gf) : \|f\|_{\text{Lip}} \leq 1, f(x^0) = 0\} \\ &\qquad + \left|\hat{\nu}(G) - \nu(G)\right| \sup\{\Psi_G(\hat{\nu})f : \|f\|_{\text{Lip}} \leq 1, f(x^0) = 0\}. \end{aligned} \tag{A.2}$$

Obviously, $|\Psi_G(\hat{\nu})f| \leq \|f\|_\infty$, and hence,

$$\sup\{\Psi_G(\hat{\nu})f : \|f\|_{\text{Lip}} \leq 1, f(x^0) = 0\} \leq \text{diam}(\mathcal{P}). \tag{A.3}$$

Also we have that

$$\begin{aligned}&\sup\{\hat{\nu}(Gf) - \nu(Gf) : \|f\|_{\text{Lip}} \leq 1, f(x^0) = 0\} \\ &= \sup_{f:\|f\|_{\text{Lip}} \leq 1, f(x^0)=0} \left\{ \|Gf\|_{\text{Lip}} \cdot \left\{ \hat{\nu}\left(\frac{Gf}{\|Gf\|_{\text{Lip}}}\right) - \nu\left(\frac{Gf}{\|Gf\|_{\text{Lip}}}\right) \right\} \right\} \\ &\leq \sup_{f:\|f\|_{\text{Lip}} \leq 1, f(x^0)=0} \big(\|G\|_\infty \cdot \|f\|_{\text{Lip}} + \|f\|_\infty \cdot \|G\|_{\text{Lip}}\big) \left\{ \hat{\nu}\left(\frac{Gf}{\|Gf\|_{\text{Lip}}}\right) - \nu\left(\frac{Gf}{\|Gf\|_{\text{Lip}}}\right) \right\} \\ &\leq \sup_{f:\|f\|_{\text{Lip}} \leq 1, f(x^0)=0} \big(\|G\|_\infty + \text{diam}(\mathcal{P}) \cdot \|G\|_{\text{Lip}}\big) \left\{ \hat{\nu}\left(\frac{Gf}{\|Gf\|_{\text{Lip}}}\right) - \nu\left(\frac{Gf}{\|Gf\|_{\text{Lip}}}\right) \right\} \\ &\leq \big(\|G\|_\infty + \text{diam}(\mathcal{P}) \cdot \|G\|_{\text{Lip}}\big) \sup_{f:\|f\|_{\text{Lip}} \leq 1, f(x^0)=0} \big\{\hat{\nu}(f) - \nu(f)\big\} \\ &= \big(\|G\|_\infty + \text{diam}(\mathcal{P}) \cdot \|G\|_{\text{Lip}}\big)\, \mathcal{W}_1(\hat{\nu}, \nu). \end{aligned} \tag{A.4}$$

From a similar but simpler reasoning,

$$|\hat{\nu}(G) - \nu(G)| \leq \|G\|_{\text{Lip}} \cdot \mathcal{W}_1(\hat{\nu}, \nu) \tag{A.5}$$

Now from (A.2), (A.3), (A.4), and (A.5), we arrive at

$$\mathcal{W}_1(\Psi_G(\hat{\nu}), \Psi_G(\nu)) \leq \left(\|G\|_\infty + 2\text{diam}(\mathcal{P}) \cdot \|G\|_{\text{Lip}}\right) \mathcal{W}_1(\hat{\nu}, \nu),$$

which is the desired inequality (A.1).

□

Lemma 2. *Let $\rho \triangleq d\xi/d\lambda^m$ denote the density of ξ w.r.t. Lebesgue measure.*

(i) For each $x \in \mathcal{P}$,

$$\left| \frac{d\tilde{\xi}^{[j+1/2]}}{d\lambda^m}(x) - \frac{d\xi}{d\lambda^m}(x) \right| \leq \|\zeta_h\|_{\text{Lip}} \, \mathcal{W}_1(\hat{\xi}^{[j]}, \xi) + h \, \|\rho\|_{\text{Lip}}.$$

(ii)

$$\|d\xi/d\xi^{[j+1/2]}\|_\infty \leq \frac{1}{(1-q)\gamma} + \frac{\lambda^m(\mathcal{P})}{q(1-\gamma)} \left(\|\zeta_h\|_{\text{Lip}} \, \mathcal{W}_1(\hat{\xi}^{[j]}, \xi) + h \, \|\rho\|_{\text{Lip}} \right)$$

for each $\gamma \in (0,1)$.

(iii) For each $x \in \mathcal{P}$,

$$\left| \nabla \left(\frac{d\tilde{\xi}^{[j+1/2]}}{d\lambda^m} \right)(x) - \nabla \left(\frac{d\xi}{d\lambda^m} \right)(x) \right| \leq m \|\nabla \zeta_h\|_{\text{Lip}} \, \mathcal{W}_1(\hat{\xi}^{[j]}, \xi) + mh \, \|\nabla \rho\|_{\text{Lip}}.$$

(iv)

$$\begin{aligned} \left\|d\xi/d\xi^{[j+1/2]}\right\|_{\text{Lip}} \leq \frac{(1-q)}{(q/\lambda^m(\mathcal{P}))^2} \Big(& \|\nabla\rho\|_\infty \cdot \left\{ \|\zeta_h\|_{\text{Lip}} \mathcal{W}_1(\hat{\xi}^{[j]}, \xi) + h \|\rho\|_{\text{Lip}} \right\} \\ & + \|\rho\|_\infty \cdot \left\{ \|\nabla\zeta_h\|_{\text{Lip}} \mathcal{W}_1(\hat{\xi}^{[j]}, \xi) + h \|\nabla\rho\|_{\text{Lip}} \right\} \Big) + \frac{\lambda^m(\mathcal{P})}{q} \|\nabla\rho\|_\infty. \end{aligned}$$

Proof. For (i), due to the construction of $\tilde{\zeta}_h$, we can use a similar argument as in Proposition 3.1 of Bolley et al. (2007). Note first that the density of $\tilde{\xi}^{[j+1/2]}$ can be written as

$$d\tilde{\xi}^{[j+1/2]}/d\lambda^m(x) = \frac{1}{N} \sum_{i=1}^{N} \tilde{\zeta}_h(x; X_i^{[j]}) = \int_{\mathcal{P}} \tilde{\zeta}_h(x; y) \hat{\xi}^{[j]}(dy),$$

and hence the difference between $d\tilde{\xi}^{[j+1/2]}/d\lambda^m(x)$ and the density $\int_{\mathcal{P}} \tilde{\zeta}_h(x; y) \xi(dy)$ obtained by smoothing ξ can be bounded as follows:

$$\begin{aligned}\left|d\tilde{\xi}^{[j+1/2]}/d\lambda^m(x) - \int_{\mathcal{P}} \tilde{\zeta}_h(x;y)\xi(dy)\right| &= \left|\int_{\mathcal{P}} \tilde{\zeta}_h(x;y)(\hat{\xi}^{[j]}(dy) - \xi(dy))\right| \\ &\leq \|\tilde{\zeta}_h(x;\cdot)\|_{\mathrm{Lip}}\, \mathcal{W}_1(\hat{\xi}^{[j]}, \xi). \qquad \text{(A.6)}\end{aligned}$$

On the other hand, due to (3.2), the distance between the smoothed density $\int_{\mathcal{P}} \tilde{\zeta}_h(x;y)\xi(dy)$ and the density $d\xi/d\lambda^m$ of ξ itself can be bounded in terms of the modulus of continuity of $d\xi/d\lambda^m$ as follows:

$$\begin{aligned}\left|\int_{\mathcal{P}} \tilde{\zeta}_h(x;y)\xi(dy) - d\xi/d\lambda^m(x)\right| &= \left|\int_{\mathcal{P}} \tilde{\zeta}_h(x;y)d\xi/d\lambda^m(y)dy - \int_{\mathcal{P}} \tilde{\zeta}_h(x;y)d\xi/d\lambda^m(x)dy\right| \\ &\leq \int_{\mathcal{P}} \tilde{\zeta}_h(x;y)\left|d\xi/d\lambda^m(y) - d\xi/d\lambda^m(x)\right|dy \\ &\leq \sup_{\substack{y\in\mathcal{P} \\ |x-y|\leq h}} |d\xi/d\lambda^m(x) - d\xi/d\lambda^m(y)| \int_{\mathcal{P}} \tilde{\zeta}_h(x;y)dy \\ &\leq h\,\|d\xi/d\lambda^m\|_{\mathrm{Lip}}. \qquad \text{(A.7)}\end{aligned}$$

Now by triangle inequality and (A.6) and (A.7), we arrive at the conclusion of (i).

Turning to (ii),

$$\begin{aligned}\|d\xi/d\xi^{[j+1/2]}\|_\infty &= \left\|\frac{d\xi/d\lambda^m}{d\xi^{[j+1/2]}/d\lambda^m}\right\|_\infty = \sup_{x\in\mathcal{P}} \frac{\rho(x)}{q/\lambda^m(\mathcal{P}) + (1-q)\frac{d\tilde{\xi}^{[j+1/2]}}{d\lambda^m}(x)} \\ &= \sup_{x\in\mathcal{P}} \frac{\rho(x)}{q/\lambda^m(\mathcal{P}) + (1-q)\frac{d\tilde{\xi}^{[j+1/2]}}{d\lambda^m}(x)} \left[\mathbb{1}_{\left\{\frac{d\tilde{\xi}^{[j+1/2]}}{d\lambda^m}(x)>\gamma\rho(x)\right\}} + \mathbb{1}_{\left\{\frac{d\tilde{\xi}^{[j+1/2]}}{d\lambda^m}(x)\leq\gamma\rho(x)\right\}}\right] \\ &\leq \sup_{x\in\mathcal{P}} \left[\frac{\rho(x)}{q/\lambda^m(\mathcal{P}) + (1-q)\gamma\rho(x)} \mathbb{1}_{\left\{\frac{d\tilde{\xi}^{[j+1/2]}}{d\lambda^m}(x)>\gamma\rho(x)\right\}} + \frac{\rho(x)}{q/\lambda^m(\mathcal{P})} \mathbb{1}_{\left\{\frac{d\tilde{\xi}^{[j+1/2]}}{d\lambda^m}(x)\leq\gamma\rho(x)\right\}}\right] \\ &\leq \frac{1}{(1-q)\gamma} + \sup_{x\in\mathcal{P}} \frac{\rho(x)}{q/\lambda^m(\mathcal{P})} \mathbb{1}_{\{\rho(x)-\|\zeta_h\|_{\mathrm{Lip}}\mathcal{W}_1(\hat{\xi}^{[j]},\xi)-h\,\|\rho\|_{\mathrm{Lip}}\leq\gamma\rho(x)\}} \\ &= \frac{1}{(1-q)\gamma} + \sup_{x\in\mathcal{P}} \frac{\rho(x)}{q/\lambda^m(\mathcal{P})} \mathbb{1}_{\left\{\rho(x)\leq\frac{\|\zeta_h\|_{\mathrm{Lip}}\mathcal{W}_1(\hat{\xi}^{[j]},\xi)+h\,\|\rho\|_{\mathrm{Lip}}}{1-\gamma}\right\}} \\ &\leq \frac{1}{(1-q)\gamma} + \frac{\|\zeta_h\|_{\mathrm{Lip}}\mathcal{W}_1(\hat{\xi}^{[j]},\xi) + h\,\|\rho\|_{\mathrm{Lip}}}{q(1-\gamma)/\lambda^m(\mathcal{P})}.\end{aligned}$$

For (iii), to bound the distance between $\partial_i(d\tilde{\xi}^{[j+1/2]}/d\lambda^m)(x)$ and $\partial_i\rho(x)$, we will consider intermediate points $\int_{\mathcal{P}} \frac{\partial}{\partial x_i}\tilde{\zeta}_h(x;y)\,\xi(dy)$ and $\int_{\tilde{\mathcal{P}}} \zeta_h(x-y)\,\partial_i\tilde{\rho}(y)dy$ where $\tilde{\mathcal{P}}$ and $\tilde{\rho}$ are slight extensions of $\tilde{\mathcal{P}}$ and ρ (precise definitions are provided below). Note first that

$$\partial_i \left(\frac{d\tilde{\xi}^{[j+1/2]}}{d\lambda^m} \right)(x) = \frac{\partial}{\partial x_i} \left(\frac{1}{N} \sum_{k=1}^{N} \tilde{\zeta}_h(x; X_k^{[j]}) \right) = \frac{1}{N} \sum_{k=1}^{N} \frac{\partial}{\partial x_i} \tilde{\zeta}_h(x; X_k^{[j]}) = \int_{\mathcal{P}} \frac{\partial}{\partial x_i} \tilde{\zeta}_h(x; y)\, \xi^{[j]}(dy),$$

and hence, as in (A.6),

$$\begin{aligned} \left| \partial_i (d\tilde{\xi}^{[j+1/2]}/d\lambda^m)(x) - \int_{\mathcal{P}} \frac{\partial}{\partial x_i} \tilde{\zeta}_h(x; y)\, \xi(dy) \right| &= \left| \int_{\mathcal{P}} \frac{\partial}{\partial x_i} \tilde{\zeta}_h(x; y)\, \hat{\xi}^{[j]}(dy) - \int_{\mathcal{P}} \frac{\partial}{\partial x_i} \tilde{\zeta}_h(x; y)\, \xi(dy) \right| \\ &\le \left\| \frac{\partial}{\partial x_i} \tilde{\zeta}_h(x; \cdot) \right\|_{\mathrm{Lip}} \mathcal{W}_1(\hat{\xi}^{[j]}, \xi) \\ &\le \| \partial_i \zeta_h \|_{\mathrm{Lip}}\, \mathcal{W}_1(\hat{\xi}^{[j]}, \xi). \end{aligned} \tag{A.8}$$

Let $\tilde{\mathcal{P}} \triangleq \prod_{i=1}^{m} \left[x_{\min}^i - h,\ x_{\max}^i + h \right]$ be the subset of $\mathbb{R}^m$ obtained by fattening $\mathcal{P}$ by h along each coordinate and $\tilde{\rho} : \tilde{\mathcal{P}} \to \mathbb{R}_+$ be the extension of ρ from $\mathcal{P}$ to $\tilde{\mathcal{P}}$ by reflection; more specifically,

$$\tilde{\rho}(x) \triangleq \rho(\tilde{x}) = \rho(\tilde{x}^1, \ldots, \tilde{x}^m) \qquad \text{where} \qquad \tilde{x}^j = \begin{cases} 2x_{\min}^j - x^j & \text{if} \quad x^j < x_{\min}^j; \\ x^j & \text{if} \quad x_{\min}^j \le x^j < x_{\max}^j; \\ 2x_{\max}^j - x^j & \text{if} \quad x_{\max}^j \le x^j \quad . \end{cases}$$

It follows from this construction that if we denote $\frac{\partial}{\partial x_i}\big(\zeta_h(x)\big)$ with $\partial_i \zeta_h(x)$,

$$\int_{\mathcal{P}} \frac{\partial}{\partial x_i} \tilde{\zeta}_h(x; y)\, \xi(dy) = \int_{\tilde{\mathcal{P}}} \partial_i \zeta_h(x - y)\, \tilde{\rho}(y) dy.$$

Note also that (obviously) $\partial_i \zeta_h(x - y) = -\frac{\partial}{\partial y_i}\big(\zeta_h(x - y)\big)$, and

$$\int_{x_{\min}^i - h}^{x_{\max}^i + h} \left(\frac{\partial}{\partial y_i} \big(\zeta_h(x - y)\big)\, \tilde{\rho}(y) + \zeta_h(x - y)\, \frac{\partial}{\partial y_i} \big(\tilde{\rho}(y)\big) \right) dy_i = \Big[\zeta_h(x - y) \tilde{\rho}(y) \Big]_{y_i = x_{\min}^i - h}^{y_i = x_{\max}^i + h} = 0$$

because $\zeta_h(x - y) = 0$ for any $x \in \mathcal{P}$ and $y \in \partial \tilde{\mathcal{P}}$ from (3.1). From these we get,

$$\begin{aligned} &\left| \int_{\mathcal{P}} \frac{\partial}{\partial x_i} \tilde{\zeta}_h(x; y)\, \xi(dy) - \int_{\tilde{\mathcal{P}}} \zeta_h(x - y)\, \partial_i \tilde{\rho}(y) dy \right| \\ &= \left| \int_{\tilde{\mathcal{P}}} \partial_i \zeta_h(x - y)\, \tilde{\rho}(y) dy - \int_{\tilde{\mathcal{P}}} \zeta_h(x - y)\, \partial_i \tilde{\rho}(y) dy \right| \\ &= \left| \int_{\tilde{\mathcal{P}}} \frac{\partial}{\partial y_i} \big(\zeta_h(x - y)\big)\, \tilde{\rho}(y) dy + \int_{\tilde{\mathcal{P}}} \zeta_h(x - y)\, \frac{\partial}{\partial y_i} \big(\tilde{\rho}(y)\big) dy \right| \\ &= 0. \end{aligned} \tag{A.9}$$

Finally,

$$\left|\int_{\tilde{\mathcal{P}}} \zeta_h(x-y)\,\partial_i\tilde{\rho}(y)dy - \partial_i\tilde{\rho}(x)\right| = \left|\int_{\tilde{\mathcal{P}}} \zeta_h(x-y)\,\partial_i\tilde{\rho}(y)dy - \int_{\tilde{\mathcal{P}}} \zeta_h(x-y)\,\partial_i\tilde{\rho}(x)dy\right|$$
$$\leq h\|\partial_i\tilde{\rho}\|_{\mathrm{Lip}} = h\|\partial_i\rho\|_{\mathrm{Lip}}. \tag{A.10}$$

Combining (A.8), (A.9), and (A.10), we arrive at (iii).

Turning to (iv), note that in general for any smooth f, g and positive constants q, v,

$$\begin{aligned}\Big\|\frac{f}{(1-q)g+q/v}\Big\|_{\mathrm{Lip}} &= \Big\|\nabla\Big(\frac{f}{(1-q)g+q/v}\Big)\Big\|_\infty = \Big\|\frac{(\nabla f)((1-q)g+q/v)-(1-q)f(\nabla g)}{((1-q)g+q/v)^2}\Big\|_\infty \\ &\leq \Big\|(1-q)\frac{(\nabla f)g - f(\nabla g)}{(q/v)^2} + \frac{(\nabla f)(q/v)}{((1-q)g+q/v)^2}\Big\|_\infty \\ &\leq \Big\|(1-q)\frac{(\nabla f)(g-f)+f(\nabla f-\nabla g)}{(q/v)^2}\Big\|_\infty + \Big\|(v/q)\nabla f\Big\|_\infty.\end{aligned}$$

Substituting f, g, v with $d\xi/d\lambda^m$, $d\xi^{[j+1/2]}/d\lambda^m$, $\lambda^m(\mathcal{P})$,

$$\begin{aligned}&\big\|d\xi/d\xi^{[j+1/2]}\big\|_{\mathrm{Lip}} \\ &\leq \frac{(1-q)}{(q/\lambda^m(\mathcal{P}))^2}\Big(\|\rho\|_{\mathrm{Lip}}\cdot\Big\|\frac{d\xi}{d\lambda^m}-\frac{d\xi^{[j+1/2]}}{d\lambda^m}\Big\|_\infty + \|\rho\|_\infty\cdot\Big\|\nabla\Big(\frac{d\xi}{d\lambda^m}\Big)-\nabla\Big(\frac{d\xi^{[j+1/2]}}{d\lambda^m}\Big)\Big\|_\infty\Big) \\ &\qquad + (\lambda^m(\mathcal{P})/q)\Big\|\nabla\Big(\frac{d\xi}{d\lambda^m}\Big)\Big\|_\infty \\ &\leq \frac{(1-q)}{(q/\lambda^m(\mathcal{P}))^2}\Big(\|\rho\|_{\mathrm{Lip}}\cdot\{\|\zeta_h\|_{\mathrm{Lip}}\mathcal{W}_1(\hat{\xi}^{[j]},\xi)+h\|\rho\|_{\mathrm{Lip}}\} + \|\rho\|_\infty\cdot\{\|\nabla\zeta_h\|_{\mathrm{Lip}}\mathcal{W}_1(\hat{\xi}^{[j]},\xi)+h\|\nabla\rho\|_{\mathrm{Lip}}\}\Big) \\ &\qquad + (\lambda^m(\mathcal{P})/q)\|\nabla\rho\|_\infty.\end{aligned}$$

This concludes the proof. □

With Lemma 1 and Lemma 2 in hand, the proof of the Proposition 2 becomes straightforward.

Proof of Proposition 2. Note that (ii) and (iv) of Lemma 2 imply that there exists a constant c_1 not depending on $\lambda^m(\mathcal{P})$ and h such that

$$\|d\xi/d\xi^{[j+1/2]}\|_\infty \leq c_1\lambda^m(\mathcal{P})\cdot\|\zeta_h\|_{\mathrm{Lip}}\cdot\mathcal{W}_1(\hat{\xi}^{[j]},\xi) + c_1(\lambda^m(\mathcal{P}))^2 + c_1h\lambda^m(\mathcal{P}) + c_1$$

and

$$\|d\xi/d\xi^{[j+1/2]}\|_{\mathrm{Lip}} \leq c_1(\lambda^m(\mathcal{P}))^2(\|\zeta_h\|_{\mathrm{Lip}}+\|\nabla\zeta_h\|_{\mathrm{Lip}})\mathcal{W}_1(\hat{\xi}^{[j]},\xi) + c_1(\lambda^m(\mathcal{P}))^2 + c_1\lambda^m(\mathcal{P}).$$

Therefore, Lemma 1 implies that

$$\begin{aligned}\mathcal{W}_1(\xi^{[j+1]},\xi) \le c_1\big(DV^2(\|\zeta_h\|_{\mathrm{Lip}}+\|\nabla\zeta_h\|_{\mathrm{Lip}})+V\|\zeta_h\|_{\mathrm{Lip}}\big)\mathcal{W}_1(\xi^{[j]},\xi)\,\mathcal{W}_1(\xi^{[j+1/2]},\hat{\xi}^{[j+1/2]})\\ + c_1\big((DV^2+V)h+(1+DV)\big)\,\mathcal{W}_1(\xi^{[j+1/2]},\hat{\xi}^{[j+1/2]})\end{aligned}$$

where c_1 is a constant that depends only on q and ξ, D denotes $\operatorname{diam}(\mathcal{P})$, and V denotes $\lambda^m(\mathcal{P})$. From Theorem 1 of Fournier and Guillin (2015), we see that $\mathbf{E}\left[\mathcal{W}_1(\xi^{[j+1/2]},\hat{\xi}^{[j+1/2]})\middle|\xi^{[j]}\right]$ can be bounded by $c_2\operatorname{diam}(\mathcal{P})\alpha(N)$ where c_2 is a constant depending only on m. Therefore,

$$\begin{aligned}\mathbf{E}\,\mathcal{W}_1(\xi^{[j+1]},\xi) &\le \mathbf{E}\left[\mathbf{E}\left[c_1\big(DV^2(\|\zeta_h\|_{\mathrm{Lip}}+\|\nabla\zeta_h\|_{\mathrm{Lip}})+V\|\zeta_h\|_{\mathrm{Lip}}\big)\mathcal{W}_1(\hat{\xi}^{[j]},\xi)\,\mathcal{W}_1(\xi^{[j+1/2]},\hat{\xi}^{[j+1/2]})\middle|\xi^{[j]}\right]\right]\\ &\quad+\mathbf{E}\left[\mathbf{E}\left[c_1\big((DV^2+V)h+(1+DV)\big)\,\mathcal{W}_1(\xi^{[j+1/2]},\hat{\xi}^{[j+1/2]})\middle|\xi^{[j]}\right]\right]\\ &\le c_1\mathbf{E}\left[\mathbf{E}\left[\mathcal{W}_1(\xi^{[j+1/2]},\hat{\xi}^{[j+1/2]})\middle|\xi^{[j]}\right]\big(DV^2(\|\zeta_h\|_{\mathrm{Lip}}+\|\nabla\zeta_h\|_{\mathrm{Lip}})+V\|\zeta_h\|_{\mathrm{Lip}}\big)\mathcal{W}_1(\hat{\xi}^{[j]},\xi)\right]\\ &\quad+c_1\mathbf{E}\left[\mathbf{E}\left[\mathcal{W}_1(\xi^{[j+1/2]},\hat{\xi}^{[j+1/2]})\middle|\xi^{[j]}\right]\big((DV^2+V)h+(1+DV)\big)\right]\\ &\le c_1c_2\alpha(N)\big(D^2V^2(\|\zeta_h\|_{\mathrm{Lip}}+\|\nabla\zeta_h\|_{\mathrm{Lip}})+DV\|\zeta_h\|_{\mathrm{Lip}}\big)\mathbf{E}\,\mathcal{W}_1(\hat{\xi}^{[j]},\xi)\\ &\quad+c_1c_2\alpha(N)\big((D^2V^2+DV)h+(D+D^2V)\big)\end{aligned}$$

Therefore, the conclusion of the proposition follows. □

Now we are ready to prove Theorem 1.

Proof of Theorem 1. Again, from Proposition 2 and Theorem 1 of Fournier and Guillin (2015), $\mathbf{E}\,\mathcal{W}_1(\hat{\xi}^{[j+1]},\xi^{[j+1]})\le cD\alpha(N)$, and hence,

$$\begin{aligned}\mathbf{E}\,\mathcal{W}_1(\hat{\xi}^{[j+1]},\xi) &\le \mathbf{E}\left[\mathcal{W}_1(\hat{\xi}^{[j+1]},\xi^{[j+1]})+\mathcal{W}_1(\xi^{[j+1]},\xi)\right]\\ &\le c\alpha(N)\big((D^2V^2+DV)h+(2D+D^2V)\big)\\ &\qquad+c\alpha(N)\big(D^2V^2(\|\zeta_h\|_{\mathrm{Lip}}+\|\nabla\zeta_h\|_{\mathrm{Lip}})+DV\|\zeta_h\|_{\mathrm{Lip}}\big)\mathbf{E}\,\mathcal{W}_1(\hat{\xi}^{[j]},\xi)\end{aligned}$$

for some c. Therefore, $w_j \triangleq \mathbf{E}\,\mathcal{W}_1(\hat{\xi}^{[j]},\xi)$ satisfies the following recursive inequality:

$$w_{j+1}\le a+bw_j$$

where $a=c\alpha(N)\big((D^2V^2+DV)h+(2D+D^2V)\big)$ and $b=c\alpha(N)\big(D^2V^2(\|\zeta_h\|_{\mathrm{Lip}}+\|\nabla\zeta_h\|_{\mathrm{Lip}})+$

$DV\|\zeta_h\|_{\mathrm{Lip}}\big)$. Solving this recursion, we get $w_j \le \frac{a}{1-b} + b^j w_0$. That is,

$$\begin{aligned}\mathbf{E}\,\mathcal{W}_1(\hat{\xi}^{[j]},\xi) \le \;& \frac{c\alpha(N)\big((D^2V^2+DV)h+(2D+D^2V)\big)}{1-c\alpha(N)\big(D^2V^2(\|\zeta_h\|_{\mathrm{Lip}}+\|\nabla\zeta_h\|_{\mathrm{Lip}})+DV\|\zeta_h\|_{\mathrm{Lip}}\big)}\\ &+\Big(c\alpha(N)\big(D^2V^2(\|\zeta_h\|_{\mathrm{Lip}}+\|\nabla\zeta_h\|_{\mathrm{Lip}})+DV\|\zeta_h\|_{\mathrm{Lip}}\big)\Big)^j\,\mathbf{E}\,\mathcal{W}_1(\hat{\xi}^{[0]},\xi).\end{aligned}$$

□

B Appendix: Consistency of Algorithm 1

In this section we provide a justification for the consistency of Algorithm 1. More specifically, let

$$\check{\xi}^{[j+1]} \triangleq \sum_{i=1}^{N} \frac{(\mu\circ f\cdot \hat{r}^m\circ f)(X_i^{[j+1/2]})}{\sum_{l=1}^{N}(\mu\circ f\cdot \hat{r}^m\circ f)(X_l^{[j+1/2]})}\delta_{X_i^{[j+1/2]}}.$$

and consider a procedure that follows the same steps 1)-4) in Section A except that

- In step 1), set, instead of (3.3),

$$\xi^{[j+1/2]} \triangleq \frac{\min\Big\{b,\; q/\lambda^m(A)+(1-q)\frac{1}{N}\sum_{i=1}^{N}\zeta_h(x-x_i')\Big\}}{\int_A \min\Big\{b,\; q/\lambda^m(A)+(1-q)\frac{1}{N}\sum_{i=1}^{N}\zeta_h(s-x_i')\Big\}\,ds}$$

- In step 4), generate samples from $\check{\xi}^{[j+1]}$ instead of $\xi^{[j+1]}$.

Then, $\hat{\xi}^{[j+1]}$ describes the samples after resampling step (in $j+1^{\text{th}}$ iteration) produced by Algorithm 1. If we set $H = (\mu\circ f)\cdot\Big(\frac{\Gamma_m\hat{r}^m\circ f}{k/N}\Big)$, $G = d\xi/d\xi^{[j+1/2]}$, $\nu = \hat{\xi}^{[j+1/2]}$, then $\Psi_H(\nu) = \check{\xi}^{[j+1]}$, $\Psi_G(\nu) = \xi^{[j+1]}$, and hence,

$$\begin{aligned}\mathcal{W}_1(\hat{\xi}^{[j+1]},\xi) &\le \mathcal{W}_1(\hat{\xi}^{[j+1]},\check{\xi}^{[j+1]}) + \mathcal{W}_1(\check{\xi}^{[j+1]},\xi^{[j+1]}) + \mathcal{W}_1(\xi^{[j+1]},\xi)\\ &= \mathcal{W}_1(\hat{\xi}^{[j+1]},\check{\xi}^{[j+1]}) + \mathcal{W}_1(\Psi_H(\nu),\Psi_G(\nu)) + \mathcal{W}_1(\xi^{[j+1]},\xi).\end{aligned}$$

Note that $\mathcal{W}_1(\hat{\xi}^{[j+1]},\check{\xi}^{[j+1]}) \le cD\alpha(N)$ from Fournier and Guillin (2015), and one can show that $\mathbf{E}\,\mathcal{W}_1(\xi^{[j+1]},\xi)$ can be bounded by $c\alpha(N)\mathbf{E}\,\mathcal{W}_1(\xi^{[j]},\xi)+d\alpha(N)$ for some c and d following a similar argument as in Proposition 2. Since H can be viewed as an approximation of G, for Algorithm 1's consistency, what is left is to show that $\mathcal{W}_1(\check{\xi}^{[j+1]},\xi^{[j+1]}) = \mathcal{W}_1(\Psi_H(\nu),\Psi_G(\nu))$ can also be bounded in a similar form by establishing some sort of modulus of continuity of $\Psi_\cdot(\nu)$ in terms of $\mathcal{W}_1$ distance w.r.t. the potential. Note first that from a straightforward algebra,

$$\big(\Psi_G(\nu) - \Psi_H(\nu)\big) f = \frac{1}{\nu(G)} \big\{ \nu\big((G-H)f\big) + \nu(H-G)\Psi_H(\nu) f \big\}$$

and hence,

$$\begin{aligned}
\mathcal{W}_1(\Psi_G(\nu), \Psi_H(\nu)) &= \sup_{f: K_f \le 1, f(x^0)=0} \big\{ \big(\Psi_G(\nu) - \Psi_H(\nu)\big) f \big\} \\
&= \frac{1}{\nu(G)} \sup_{f: K_f \le 1,, f(x^0)=0} \big\{ \nu\big((G-H)f\big) + \nu(H-G)\Psi_H(\nu) f \big\} \qquad \text{(B.1)} \\
&\le \frac{1}{\nu(G)} \sup_{f: K_f \le 1, f(x^0)=0} \big\{ \nu\big(|H-G| \, \|f\|_\infty\big) + |\nu(H-G)| \, \|f\|_\infty \big\} \\
&\le \frac{1}{\nu(G)} \big\{ \nu\big(|H-G|\big) \operatorname{diam}(\mathcal{P}) + |\nu(H-G)| \operatorname{diam}(\mathcal{P}) \big\} \\
&\le \frac{2 \operatorname{diam}(\mathcal{P})}{\nu(G)} \nu(|H-G|). \qquad \text{(B.2)}
\end{aligned}$$

Therefore,

$$\begin{aligned}
\mathcal{W}_1(\check{\xi}^{[j+1]}, \xi^{[j+1]}) &\le \frac{2 \operatorname{diam}(\mathcal{P})}{\hat{\xi}^{[j+1/2]}(d\xi/d\xi^{[j+1/2]})} \, \hat{\xi}^{[j+1/2]} \left(\left| \mu \circ f \cdot \frac{\Gamma_m}{k/N} \hat{r}^m \circ f - d\xi/d\xi^{[j+1/2]} \right| \right) \\
&\le 2\, b \operatorname{diam}(\mathcal{P}) \, \hat{\xi}^{[j+1/2]} \left(\left| \mu \circ f \cdot \frac{\Gamma_m}{k/N} \hat{r}^m \circ f - d\xi/d\xi^{[j+1/2]} \right| \right).
\end{aligned}$$

where the second inequality is from the construction of $\xi^{[j+1/2]}$. Lemma 3 provides the desired bound for the RHS. For Lemma 3, we make a few additional assumptions.

A3. The target density μ is bounded away from above and below;

A4. Spectrum of Df is bounded away from 0 and $\pm\infty$;

A5. There exists a constant $c_m > 0$ and $\delta_0 > 0$ such that if $y_0 \in \mathcal{M}$

$$\int_{B(y_0;\delta_1) \setminus B(y_0;\delta_2)} \mathcal{H}^m(dy) \ge c_m \left(\delta_1^m - \delta_2^m \right) \qquad \text{(B.3)}$$

for $\delta_0 \ge \delta_1 \ge \delta_2 \ge 0$.

Lemma 3. *For any given $\epsilon > 0$, one can choose k as a function of N so that there exists c such that*

$$\mathbf{E}\, \hat{\xi}_N^{[j+1/2]} \left(\left| \mu \circ f \cdot \frac{\Gamma_m}{k/N} \hat{r}^m \circ f - d\xi/d\xi^{[j+1/2]} \right| \right) \le c\big(1 + \|\nabla \zeta_h\|_{\mathrm{Lip}} \mathcal{W}_1(\xi^{[j]}, \xi)\big) N^{-\frac{1-\epsilon}{2(m+1)}}. \qquad \text{(B.4)}$$

Proof. First note that from (2.3),

$$\begin{aligned}\left|\mu\circ f\cdot\frac{\Gamma_m}{k/N}\hat{r}^m\circ f-d\xi/d\xi^{[j+1/2]}\right|&=|\mu\circ f|\cdot\left|\frac{\Gamma_m}{k/N}\hat{r}^m\circ f-\frac{J_m f}{d\xi^{[j+1/2]}/d\lambda^m}\right|\\&=|\mu\circ f|\cdot\left|\frac{\Gamma_m}{k/N}\hat{r}^m\circ f-\frac{1}{p_Y\circ f}\right|\end{aligned}$$

where p_Y is the density of $f(X_i^{[j+1/2]})$'s w.r.t. the Hausdorff measure. Now we consider the following decomposition:

$$\begin{aligned}|\mu\circ f|\cdot\left|\frac{\Gamma_m}{k/N}\hat{r}^m\circ f-\frac{1}{p_Y\circ f}\right|&\le|\mu\circ f|\cdot\left|\frac{\Gamma_m}{k/N}\hat{r}^m\circ f-\frac{\Gamma_m}{k/N}r_k^m\circ f\right|+|\mu\circ f|\cdot\left|\frac{\Gamma_m}{k/N}r_k^m\circ f-\frac{1}{p_Y\circ f}\right|\\&=(I)+(II)\end{aligned}\tag{B.5}$$

where for each y_0, $r_k(y_0)$ is the real number such that

$$\int_{B(y_0;r_k(y_0))}p_Y(y)\mathcal{H}^m(dy)=k/N.$$

For (I),

$$\begin{aligned}&\mathbf{E}\,\frac{\hat{\xi}_N^{[j+1/2]}(|\mu\circ f|\cdot|\hat{r}^m\circ f-r_k^m\circ f|)}{k/N}=\mathbf{E}\,\frac{\left|\mu(f(X_i^{[j+1/2]}))\right|\cdot\left|\hat{r}^m(f(X_1^{[j+1/2]}))-r_k^m(f(X_1^{[j+1/2]}))\right|}{k/N}\\&=\mathbf{E}\left[\mathbf{E}\left[\left.\frac{\left|\mu(f(X_1^{[j+1/2]}))\right|\cdot\left|\hat{r}^m(f(X_1^{[j+1/2]}))-r_k^m(f(X_1^{[j+1/2]}))\right|}{k/N}\right|f(X_1^{[j+1/2]})\right]\right].\end{aligned}$$

We first study the inner conditional expectation given $f(X_1^{[j+1/2]})=y_0$, which will be denoted by $\mathbf{E}_{y_0}$ from now on, i.e.,

$$\mathbf{E}_{y_0}\frac{|\mu(y_0)|\cdot|\hat{r}^m(y_0)-r_k^m(y_0)|}{k/N}\triangleq\mathbf{E}\left[\left.\frac{\left|\mu(f(X_1^{[j+1/2]}))\right|\cdot\left|\hat{r}^m(f(X_1^{[j+1/2]}))-r_k^m(f(X_1^{[j+1/2]}))\right|}{k/N}\right|f(X_1^{[j+1/2]})=y_0\right].$$

With this notation,

$$
\begin{aligned}
&\mathbf{E}_{y_0} \frac{|\mu(y_0)| \cdot |\hat{r}^m(y_0) - r_k^m(y_0)|}{k/N} = \int_0^\infty \mathbf{P}_{y_0}\left(\frac{|\mu(y_0)| \cdot |\hat{r}^m(y_0) - r_k^m(y_0)|}{k/N} \geq s \right) ds \\
&\leq \gamma + \int_\gamma^\infty \mathbf{P}_{y_0}\left(\frac{|\mu(y_0)| \cdot |\hat{r}^m(y_0) - r_k^m(y_0)|}{k/N} \geq s \right) ds \\
&\leq \gamma + \int_\gamma^\infty \mathbf{P}_{y_0}\left(\hat{r}^m(y_0) - r_k^m(y_0) \geq \frac{ks}{N\mu(y_0)} \right) ds + \int_\gamma^\infty \mathbf{P}_{y_0}\left(r_k^m(y_0) - \hat{r}^m(y_0) \geq \frac{ks}{N\mu(y_0)} \right) ds
\end{aligned}
\tag{B.6}
$$

where $\mathbf{P}_{y_0}$ is the corresponding conditional probability given $f(X_1^{[j+1/2]}) = y_0$. Note that since $\mathcal{M}$ is bounded, the upper limit of the first integral is in fact finite:

$$
\int_\gamma^\infty \mathbf{P}_{y_0}\left(\hat{r}^m(y_0) - r_k^m(y_0) \geq \frac{ks}{N\mu(y_0)} \right) ds = \int_\gamma^{(N/k)\mathrm{diam}(\mathcal{M})} \mathbf{P}_{y_0}\left(\hat{r}^m(y_0) - r_k^m(y_0) \geq \frac{ks}{N\mu(y_0)} \right) ds.
$$

Also,

$$
\begin{aligned}
&\mathbf{P}_{y_0}\left(\hat{r}^m(y_0) - r_k^m(y_0) \geq \frac{ks}{N\mu(y_0)} \right) \\
&= \mathbf{P}_{y_0}\left(\text{less than } k \text{ points fall inside } B\Big(y_0; (r_k^m(y_0) + \mu(y_0)^{-1} ks/N)^{1/m}\Big) \right) \\
&= \mathbf{P}(\mathrm{Bin}(N, q) \leq k)
\end{aligned}
\tag{B.7}
$$

where $q = q(s, y_0) \triangleq \int_{B(y_0;(r_k^m(y_0)+\mu(y_0)^{-1}ks/N)^{1/m})} \frac{\frac{d\xi_N^{[j+1/2]}}{d\lambda^m} \circ f^{-1}(y)}{J_m f \circ f^{-1}(y)} \mathcal{H}^m(dy)$. Note that

$$
\begin{aligned}
q - k/N &= \int_{B(y_0;(r_k^m(y_0)+\mu(y_0)^{-1}kx/N)^{1/m})} \frac{\frac{d\xi_N^{[j+1/2]}}{d\lambda^m} \circ f^{-1}(y)}{J_m f \circ f^{-1}(y)} \mathcal{H}^m(dy) - k/N \\
&= \int_{B(y_0;(r_k^m(y_0)+\mu(y_0)^{-1}ks/N)^{1/m})} p_Y(y) \mathcal{H}^m(dy) - \int_{B(y_0;r_k(y_0))} p_Y(y) \mathcal{H}^m(dy) \\
&= \int_{B(y_0;(r_k^m(y_0)+\mu(y_0)^{-1}ks/N)^{1/m}) \setminus B(y_0;r_k(y_0))} p_Y(y) \mathcal{H}^m(dy) \\
&\geq \frac{csk}{N} \cdot \frac{p_Y(y_0 + h')}{\mu(y_0)}
\end{aligned}
$$

where $y_0 + h' \in B(y_0; (r_k^m(y_0) + \mu(y_0)^{-1} ks/N)^{1/m})$ by (B.3) and the mean-value theorem as far as

$(\frac{ks}{N\mu(y_0)} + r_k^m)^{1/m} \leq \delta_0$. Therefore, from Hoeffding's inequality and (B.7),

$$\mathbf{P}_{y_0}\left(\hat{r}^m(y_0) - r_k^m(y_0) \geq \frac{ks}{N\mu(y_0)}\right) \leq \exp\left(-2N\left(\frac{cks\, p_Y(y_0+h')}{N\mu(y_0)}\right)^2\right)$$

for $s \leq (N/k)\mu(y_0)(\delta_0^m - r_k^m(y_0))$. In view of this,

$$\begin{aligned}
&\int_\gamma^{(N/k)\mu(y_0)\mathrm{diam}(\mathcal{M})^m} \mathbf{P}_{y_0}\left(\hat{r}^m(y_0) - r_k^m(y_0) \geq \frac{ks}{N\mu(y_0)}\right) ds \\
&\leq \int_\gamma^{(N/k)\mu(y_0)\mathrm{diam}(\mathcal{M})^m} \mathbf{P}_{y_0}\left(\hat{r}^m(y_0) - r_k^m(y_0) \geq \frac{k\gamma}{N\mu(y_0)}\right) ds \\
&\leq (N/k)\mu(y_0)\mathrm{diam}(\mathcal{M})^m \exp\left(-\frac{2c^2k^2\gamma^2\, p_Y(y_0+h')^2}{N\mu(y_0)^2}\right)
\end{aligned}$$

if $\gamma \leq (N/k)(\delta_0^m - r_k^m(y_0))$. Similarly, the second integral in (B.6) can be bounded by first noting that $\hat{r}$ is non-negative (and hence the upper limit of the integral is finite), and then applying Hoeffding's inequality:

$$\begin{aligned}
\int_\gamma^\infty \mathbf{P}_{y_0}\left(r_k^m(y_0) - \hat{r}^m(y_0) \geq \frac{ks}{N\mu(y_0)}\right) ds &\leq \int_\gamma^{(N/k)\mu(y_0)r_k(y_0)^m} \mathbf{P}_{y_0}\left(r_k^m(y_0) - \hat{r}^m(y_0) \geq \frac{k\gamma}{N\mu(y_0)}\right) ds \\
&\leq (N/k)\mu(y_0){r_k}^m(y_0) \exp\left(-\frac{2c^2k^2\gamma^2\, p_Y(y_0+h')^2}{N\mu(y_0)^2}\right)
\end{aligned}$$

for $\gamma \leq (N/k)(\delta_0^m - {r_k}^m(y_0))$. From these along with (B.6), we conclude that

$$\begin{aligned}
\mathbf{E}_{y_0}\frac{|\mu(y_0)|\cdot|\hat{r}^m(y_0) - r_k^m(y_0)|}{k/N} &\leq \gamma + (N/k)\mu(y_0)\big(\mathrm{diam}(\mathcal{M})^m + r_k^m(y_0)\big)\exp\left(-\frac{2c^2k^2\gamma^2\, p_Y(y_0+h')^2}{N\mu(y_0)^2}\right) \\
&\leq \gamma + 2(N/k)\mu(y_0)\mathrm{diam}(\mathcal{M})^m \exp\left(-\frac{2c^2k^2\gamma^2\, p_Y(y_0+h')^2}{N\mu(y_0)^2}\right).
\end{aligned}$$

for $\gamma \leq (N/k)(\delta_0^m - {r_k}^m(y_0))$. Choosing $\gamma = \frac{N^{1/2+\beta}}{k}\frac{\mu(y_0)}{p_Y(y_0)}$ and $k = N^\alpha$ for $\alpha \in (1/2, 1)$ and $\beta \in (0, 1/2)$, we get

$$|h'| \leq \left(r_k^m(y_0) + \frac{k\gamma/N}{\mu(y_0)}\right)^{1/m} = \left(r_k^m(y_0) + \frac{N^{\beta-1/2}}{p_Y(y_0)}\right)^{1/m}$$

and

$$\begin{aligned}
&\mathbf{E}_{y_0} \frac{|\mu(y_0)| \cdot |\hat{r}^m(y_0) - r_k^m(y_0)|}{k/N} \\
&\leq N^{1/2+\beta-\alpha} \frac{\mu(y_0)}{p_Y(y_0)} + 2N^{1-\alpha}\mu(y_0)\mathrm{diam}(\mathcal{M})^m \exp\left(-2c^2 N^{2\beta} \left(\frac{p_Y(y_0+h')}{p_Y(y_0)}\right)^2\right) \\
&\leq N^{1/2+\beta-\alpha} \frac{\mu(y_0)}{p_Y(y_0)} + 2N^{1-\alpha}\mu(y_0)\mathrm{diam}(\mathcal{M})^m \exp\left(-2c^2 N^{2\beta} \left(\underline{p_Y}/\overline{p_Y}\right)^2\right). \qquad \text{(B.8)}
\end{aligned}$$

Therefore,

$$\mathbf{E} \frac{\hat{\xi}_N^{[j+1/2]}(|\mu \circ f| \cdot |\Gamma_m \hat{r}^m \circ f - \Gamma_m r_k^m \circ f|)}{k/N} \leq c\{\underline{p_Y}^{-1} N^{1/2+\beta-\alpha} + N^{1-\alpha} \exp(-cN^{2\beta}(\underline{p_Y}/\overline{p_Y})^2\} \tag{B.9}$$

for some $c > 0$ that depends only on f, μ, and $\mathrm{diam}(M)$.

Now, turning to (II) of (B.5), we first prove that for y_0 and δ such that $\bar{f}^{-1}(B(y_0;\delta)) \subseteq A$ (where $\bar{f}$ is defined in (B.11)),

$$\left| \frac{1}{\Gamma_m \delta^m} \int_{B(y_0;\delta)} \mathcal{H}^m(dy) - 1 \right| \leq c_f' \delta \tag{B.10}$$

for c_f' that depends only on f. Let $\bar{f}$ be the linear approximation of f at $x_0 = f^{-1}(y_0)$, i.e.,

$$\bar{f}(x_0 + h) = f(x_0) + Df(x_0)h, \tag{B.11}$$

and $\bar{\mathcal{H}}^m$ be the Hausdorff measure on $\bar{f}(\mathcal{P})$. Then,

$$\begin{aligned}
\int_{B(y_0;\delta)} \mathcal{H}^m(dy) - \Gamma_m \delta^m &= \int_{B(y_0;\delta)} \mathcal{H}^m(dy) - \int_{B(y_0;\delta)} \bar{\mathcal{H}}^m(dy) \\
&= \int_{f^{-1}(B(y_0;\delta))} J_m f(x) \lambda^m(dx) - \int_{\bar{f}^{-1}(B(y_0;\delta))} J_m \bar{f}(x) \lambda^m(dx) \\
&= \int_{f^{-1}(B(y_0;\delta)) \cap \bar{f}^{-1}(B(y_0;\delta))} \left(J_m f(x) - J_m \bar{f}(x)\right) \lambda^m(dx) \\
&\quad + \int_{f^{-1}(B(y_0;\delta)) \setminus \bar{f}^{-1}(B(y_0;\delta))} J_m f(x) \lambda^m(dx) \\
&\quad - \int_{\bar{f}^{-1}(B(y_0;\delta)) \setminus f^{-1}(B(y_0;\delta))} J_m \bar{f}(x) \lambda^m(dx) \\
&= (\mathrm{I})' + (\mathrm{II})' - (\mathrm{III})'
\end{aligned}$$

Since $J_m\bar{f}(x) = J_m f(x_0)$, the first term (I)$'$ in the previous display can be bounded as follows:

$$
\begin{aligned}
|(\mathrm{I})'| &= \int_{f^{-1}(B(y_0;\delta))\cap\bar{f}^{-1}(B(y_0;\delta))} \big|J_m f(x) - J_m f(x_0)\big| \lambda^m(dx) \\
&\le \int_{\bar{f}^{-1}(B(y_0;\delta))} J_m\bar{f}(x)\lambda^m(dx) \sup_{x\in\bar{f}^{-1}(B(y_0;\delta))} \left|1 - \frac{J_m f(x)}{J_m f(x_0)}\right| \\
&\le \Gamma_m\delta^m \cdot \sup_{x\in\bar{f}^{-1}(B(y_0;\delta))} \left|1 - \frac{J_m f(x)}{J_m f(x_0)}\right|
\end{aligned}
$$

To get a bound for (II)$'$, note that

$$
\begin{aligned}
|(\mathrm{II})'| &\le \int_{f^{-1}(B(y_0;\delta))\backslash\bar{f}^{-1}(B(y_0;\delta))} J_m f(x_0)\lambda^m(dx) \cdot \sup_{x\in\bar{f}^{-1}(B(y_0;\delta))} \left|\frac{J_m f(x)}{J_m f(x_0)}\right| \\
&= \int_{f\left(f^{-1}(B(y_0;\delta))\backslash\bar{f}^{-1}(B(y_0;\delta))\right)} \bar{\mathcal{H}}^m(dy) \cdot \sup_{x\in\bar{f}^{-1}(B(y_0;\delta))} \left|\frac{J_m f(x)}{J_m f(x_0)}\right|
\end{aligned}
$$

To bound the integral, we prove that $f\big(f^{-1}(B(y_0;\delta))\backslash\bar{f}^{-1}(B(y_0;\delta))\big)$ is close to the boundary of $B(y_0;\delta)$—i.e., $f\big(f^{-1}(B(y_0;\delta))\backslash\bar{f}^{-1}(B(y_0;\delta))\big) \subset B(y_0;\delta)\backslash B(y_0;\delta-\epsilon(\delta))$ where $\epsilon(\delta) = o(\delta)$ as $\delta \to 0$. Suppose that $y \in f\big(f^{-1}(B(y_0;\delta))\backslash\bar{f}^{-1}(B(y_0;\delta))\big)$. Then, there exists an $h \in \mathbb{R}^m$ such that $x_0 + h = f^{-1}(y)$, $f(x_0+h) \in B(y_0;\delta)$, and $\bar{f}(x_0+h) \notin B(y_0;\delta)$.

Since f is C^2, the k^{th} component of f can be written as

$$
f_k(x_0+h) = f_k(x_0) + Df_k(x_0)h + \frac{1}{2}h^T R_k(x_0+h)h
$$

where $R_k(x_0+h) = \int_0^1 (1-t)D^2 f_k(x_0+th)dt$. Therefore, $\|R_k(x_0+h)\| \le \sup_{t\in[0,1]} \|D^2 f_k(x_0+th)\|$,

$$
\|f(x_0+h) - \bar{f}(x_0+h)\|_2 \le \frac{1}{2}\sum_{k=1}^{n} |h^T R_k(x_0+h)h| \le \frac{1}{2}\sup_{x\in\mathcal{P}} \|D^2 f_k(x)\| \|h\|_2^2. \tag{B.12}
$$

Assumption A4 guarantees that $\|f^{-1}\|_{\text{Lip}} > 0$ and $\operatorname{diam}(f^{-1}(B(y_0;\delta)) < 2\|f^{-1}\|_{\text{Lip}}\delta$, and since $x_0 + h \in f^{-1}(B(y_0;\delta))$—i.e., $\|h\|_2 \le 2\|f^{-1}\|_{\text{Lip}}\delta$—(B.12) becomes

$$
\|f(x_0+h) - \bar{f}(x_0+h)\|_2 \le 2\|f^{-1}\|_{\text{Lip}}^2 \sup_{x\in\mathcal{P}} \|D^2 f_k(x)\|\delta^2 \triangleq c_f\,\delta^2.
$$

Since $\tilde{f}(x_0+h) \notin B(y_0,\delta)$, the above inequality implies that $f(x_0+h) \in B(y_0;\delta)\backslash B(y_0;\delta - c_f\delta^2)$, and hence, $f\big(f^{-1}(B(y_0;\delta))\backslash\bar{f}^{-1}(B(y_0;\delta))\big)$ is a subset of $B(y_0;\delta)\backslash B(y_0;\delta - c_f\delta^2)$. Now, we can bound the integral in (II)$'$

$$\int_{f\left(f^{-1}(B(y_0;\delta))\setminus\bar{f}^{-1}(B(y_0;\delta))\right)} \bar{\mathcal{H}}^m(dy) \leq \Gamma_m(\delta^m - (\delta - c_f\delta^2)^m) \leq m\Gamma_m c_f \delta^{m+1},$$

which in turn implies

$$(\text{II})' < m\Gamma_m c_f \delta^{m+1} \cdot \sup_{x\in\bar{f}^{-1}(B(y_0;\delta))} \left|\frac{J_m f(x)}{J_m f(x_0)}\right|.$$

The following bound for (III)$'$ can be obtained by an essentially identical (but only simpler) argument:

$$(\text{III})' = \int_{\bar{f}^{-1}(B(y_0;\delta))\setminus f^{-1}(B(y_0;\delta))} J_m\bar{f}(x)\lambda^m(dx) \leq m\Gamma_m c_f \delta^{m+1}$$

Now, combining the bounds for (I)$'$, (II)$'$, and (III)$'$, we arrive at

$$\left|\frac{1}{\Gamma_m\delta^m}\int_{B(y_0;\delta)} \mathcal{H}^m(dy) - 1\right| \leq \sup_{x\in\bar{f}^{-1}(B(y_0;\delta))}\left|1 - \frac{J_m f(x)}{J_m f(x_0)}\right| + mc_f\delta\left(\sup_{x\in\bar{f}^{-1}(B(y_0;\delta))}\left|\frac{J_m f(x)}{J_m f(x_0)}\right| + 1\right) \tag{B.13}$$

and

$$\sup_{x\in\bar{f}^{-1}(B(y_0;\delta))}\left|1 - \frac{J_m f(x)}{J_m f(x_0)}\right| \leq \delta\|J_m f\|_{\text{Lip}}\,\|f^{-1}\|_{\text{Lip}}/\underline{J_m f}$$

where $\underline{J_m f} = \inf_{x\in\mathcal{P}} J_m f(x)$. Letting $c_f' = \|J_m f\|_{\text{Lip}}\|f^{-1}\|_{\text{Lip}}/\underline{J_m f} + mc_f\left(\sup_{x\in\bar{f}^{-1}(B(y_0;\delta))}\left|\frac{J_m f(x)}{J_m f(x_0)}\right| + 1\right)$, we arrive at (B.10). Note that from the mean value theorem,

$$\begin{aligned}\int_{B(y_0;\delta)} p_Y(y)\mathcal{H}^m(dy) &= \int_{f^{-1}(B(y_0;\delta))} J_m f(x) p_Y(f(x))\lambda^m(dx) \\ &= \int_{f^{-1}(B(f(x_0);\delta))} J_m f(x)\lambda^m(dx)\; p_Y \circ f(x_0 + h^*) \\ &= \int_{B(y_0;\delta)} \mathcal{H}^m(dy)\; p_Y \circ f(x_0 + h^*)\end{aligned} \tag{B.14}$$

for some h^* such that $f(x_0 + h^*) \in B(y_0;\delta)$ where $x_0 = f^{-1}(y_0)$. Substituting r_k for δ in (B.14) and using the definition of $r_k(y_0)$, we get

$$\frac{\Gamma_m r_k^m \circ f(x_0)}{k/N} = \frac{\Gamma_m r_k^m \circ f(x_0)}{\int_{B(f(x_0);r_k\circ f(x_0))} \mathcal{H}^m(dy)} \frac{p_Y \circ f(x_0)}{p_Y \circ f(x_0 + h^*)} \frac{1}{p_Y \circ f(x_0)}. \tag{B.15}$$

We therefore arrive at the bound

$$\begin{aligned}
&\left|\frac{1}{p_Y\circ f(x_0)} - \frac{\Gamma_m r_k^m \circ f(x_0)}{k/N}\right| \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad\,\,\text{(B.16)}\\
&= \frac{1}{p_Y(y_0)}\left(\frac{p_Y\circ f(x_0+h^*) - p_Y\circ f(x_0)}{p_Y\circ f(x_0+h^*)} + \frac{p_Y\circ f(x_0)}{p_Y\circ f(x_0+h^*)}\frac{\Gamma_m r_k^m(y_0)}{\int_{B(y_0;r_k(y_0))}\mathcal{H}^m(dy)}\left(\frac{\int_{B(y_0;r_k(y_0))}\mathcal{H}^m(dy)}{\Gamma_m r_k^m(y_0)} - 1\right)\right) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad\,\,\text{(B.17)}
\end{aligned}$$

for y_0 s.t. $\bar{f}^{-1}(B(y_0;\delta)) \subseteq \mathcal{P}$ and h^* such that $f(x_0+h^*) \in B(y_0;\delta)$. Note that if $x_0 \in \mathcal{P}^{-r_k\circ f(x_0)\|f^{-1}\|_{\mathrm{Lip}}}$, then $\bar{f}^{-1}(B(f(x_0);r_k\circ f(x_0)) \subseteq \mathcal{P}$. If, in addition, $x_0 \in \mathcal{P}^{-2r_k\circ f(x_0)\|f^{-1}\|_{\mathrm{Lip}}}$, then $x_0+h^* \in f^{-1}(B(f(x_0);r_k\circ f(x_0))$ implies $|h^*| \le r_k\circ f(x_0)\|f^{-1}\|_{\mathrm{Lip}}$, and hence, $x_0+h^* \in \mathcal{P}^{-r_k\circ f(x_0)\|f^{-1}\|_{\mathrm{Lip}}}$. For notational simplicity, let $\bar{r}_k \triangleq \|r_k\|_\infty$, $\underline{p_Y} \triangleq \|1/p_Y\|_\infty$, and $\overline{p_Y} \triangleq \|p_Y\|_\infty$. Then, if $x_0 \in \mathcal{P}^{-2\bar{r}_k\|f^{-1}\|_{\mathrm{Lip}}}$, then assuming that $\bar{r}_k$ is sufficiently small,

$$\begin{aligned}
\left|\frac{1}{p_Y\circ f(x_0)} - \frac{\Gamma_m r_k^m\circ f(x_0)}{k/N}\right| &\le \frac{1}{p_Y(y_0)}\left(\frac{\|p_Y\circ f\|_{\mathrm{Lip}}h^*}{p_Y\circ f(x_0+h^*)} + \frac{p_Y(y_0)}{p_Y(y_0+h^*)}\frac{c'_f r_k(y_0)}{1-c'_f r_k(y_0)}\right)\\
&\le \frac{r_k(y_0)}{p_Y(y_0)}\left(\frac{\|f^{-1}\|_{\mathrm{Lip}}\|p_Y\circ f\|_{\mathrm{Lip}}}{\underline{p_Y}} + \frac{\overline{p_Y}}{\underline{p_Y}}\frac{c'_f}{1-c'_f r_k(y_0)}\right)\\
&\le \frac{c\bar{r}_k}{p_Y(y_0)}\left(\|p_Y\circ f\|_{\mathrm{Lip}} + 1\right).
\end{aligned}$$

for some $c > 0$ that depends only on f, b, and q. From Lemma 2 and the construction of $d\xi^{[j+1/2]}$,

$$\begin{aligned}
\|p_Y\circ f\|_{\mathrm{Lip}} &\le \|d\xi^{[j+1/2]}/d\lambda^m\|_{\mathrm{Lip}}\cdot\|1/J_m f\|_\infty + \|1/J_m f\|_{\mathrm{Lip}}\,\|d\xi^{[j+1/2]}/d\lambda^m\|_\infty\\
&\le c\big(1 + h + \|\nabla\zeta_h\|_{\mathrm{Lip}}\mathcal{W}_1(\xi^{[j]},\xi)\big).
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\mathbf{E}\,\hat{\xi}^{[j+1/2]}\left(|\mu\circ f|\cdot\left|\frac{\Gamma_m r_k^m\circ f}{k/N}-\frac{1}{p_Y\circ f}\right|\right)\\
&=\mathbf{E}\,\mu(f(X_1^{[j+1/2]}))\left|\frac{1}{p_Y(f(X_1^{[j+1/2]}))}-\frac{\Gamma_m r_k^m(f(X_1^{[j+1/2]}))}{k/N}\right|\\
&\qquad\qquad\qquad\cdot\left(\mathbb{1}_{\{X_1^{[j+1/2]}\in\mathcal{P}\setminus\mathcal{P}^{-2\bar{r}_k\|f^{-1}\|_{\mathrm{Lip}}}\}}+\mathbb{1}_{\{X_1^{[j+1/2]}\in\mathcal{P}^{-2\bar{r}_k\|f^{-1}\|_{\mathrm{Lip}}}\}}\right)\\
&\le\|\mu\|_\infty\Big(\frac{\Gamma_m\bar{r}_k^m}{k/N}+\underline{p_Y}^{-1}\Big)\mathbf{P}\big(X_N^{[i]}\in\mathcal{P}\setminus\mathcal{P}^{-2\bar{r}_k\|f^{-1}\|_{\mathrm{Lip}}}\big)\\
&\qquad+\mathbf{E}\,\mu(f(X_1^{[j+1/2]}))\left|\frac{1}{p_Y(f(X_1^{[j+1/2]}))}-\frac{\Gamma_m r_k^m(f(X_1^{[j+1/2]}))}{k/N}\right|\mathbb{1}_{\{X_1^{[j+1/2]}\in\mathcal{P}^{-2\bar{r}_k\|f^{-1}\|_{\mathrm{Lip}}}\}}\\
&\le\|\mu\|_\infty\Big(\frac{\Gamma_m\bar{r}_k^m}{k/N}+\underline{p_Y}^{-1}\Big)\mathbf{P}(X_N^{[i]}\in\mathcal{P}\setminus\mathcal{P}^{-2\bar{r}_k\|f^{-1}\|_{\mathrm{Lip}}})+c\mathbf{E}\,\frac{\bar{r}_k}{p_Y(f(X_1^{[j+1/2]}))}\left(\|p_Y\circ f\|_{\mathrm{Lip}}+1\right)\\
&\le c\big(1+h+\|\nabla\zeta_h\|_{\mathrm{Lip}}\mathcal{W}_1(\xi^{[j]},\xi)\big)\bar{r}_k.
\end{aligned}$$

for some (new) constant $c > 0$. Therefore, together with (B.9) and noting that $\bar{r}_k^m = O((k/N)) = O(N^{\alpha-1})$ (since p_Y is bounded from below by construction),

$$\begin{aligned}
&\mathbf{E}\,\hat{\xi}^{[j+1/2]}\left(\left|\mu\circ f\cdot\frac{\Gamma_m}{k/N}\hat{r}^m\circ f-d\xi/d\xi^{[j+1/2]}\right|\right)\\
&\le c\Big\{\big(1+\|\nabla\zeta_h\|_{\mathrm{Lip}}\mathcal{W}_1(\xi^{[j]},\xi)\big)N^{\frac{\alpha-1}{m}}+N^{1/2+\beta-\alpha}+N^{1-\alpha}\exp(-cN^{2\beta}(\underline{p_Y}/\overline{p_Y})^2\Big\}
\end{aligned}$$

Noting that $\underline{p_Y}/\overline{p_Y}$ is bounded from below by construction, we see that for any $\epsilon > 0$, by considering β small enough and $\alpha \approx \frac{m+2}{2(m+1)}$, one can always find a (new) constant $c > 0$ such that

$$\mathbf{E}\,\hat{\xi}_N^{[j+1/2]}\left(\left|\mu\circ f\cdot\frac{\Gamma_m}{k/N}\hat{r}^m\circ f-d\xi/d\xi_N^{[j+1/2]}\right|\right)\le c\big(1+\|\nabla\zeta_h\|_{\mathrm{Lip}}\mathcal{W}_1(\xi^{[j]},\xi)\big)N^{-\frac{1-\epsilon}{2(m+1)}}.$$

□

C Appendix: Details of the MH Algorithm in Figure 2

To produce Figure 2, we ran random walk MH algorithm in $\mathcal{P} = [0, 100] \times [0, 100]$ with the standard normal increments and starting point $(\theta_1, \theta_2) = (50, 50)$ at the center of $\mathcal{P}$. For a proposal (θ_1^*, θ_2^*) from the current state (θ_1, θ_2), the acceptance probability was $J_2f(\theta_1^*, \theta_2^*)/J_2f(\theta_1, \theta_2)$, where J_2f is as defined in (4.2). To mitigate the irregularity along the boundaries of $\mathcal{P}$, if the proposal (θ_1^*, θ_2^*) falls outside of $\mathcal{P}$, we reflected the proposal along the boundaries.

References

Asmussen, S. and Glynn, P. W. (2007). *Stochastic simulation: algorithms and analysis*, volume 57. Springer Science & Business Media.

Boender, C., Caron, R. J., McDonald, J., Kan, A. R., Romeijn, H. E., Smith, R. L., Telgen, J., and Vorst, A. (1991). Shake-and-bake algorithms for generating uniform points on the boundary of bounded polyhedra. *Operations research*, 39(6):945–954.

Bolley, F., Guillin, A., and Villani, C. (2007). Quantitative concentration inequalities for empirical measures on non-compact spaces. *Probability Theory and Related Fields*, 137(3-4):541–593.

Boneh, A. and Golan, A. (1979). Constraints' redundancy and feasible region boundedness by random feasible point generator (rfpg). In *Third European congress on operations research (EURO III), Amsterdam*.

Bouchard-Côté, A., Vollmer, S. J., and Doucet, A. (2018). The bouncy particle sampler: A non-reversible rejection-free markov chain monte carlo method. *Journal of the American Statistical Association*, 113(522):855–867.

Diaconis, P., Holmes, S., and Shahshahani, M. (2013). Sampling from a manifold. In Jones, G. and Shen, X., editors, *Advances in Modern Statistical Theory and Applications: A Festschrift in Honor of Morris L. Eaton*, pages 102–125. Institute of Mathematical Statistics.

Dieker, A. and Vempala, S. S. (2015). Stochastic billiards for sampling from the boundary of a convex set. *Mathematics of Operations Research*, 40(4):888–901.

Federer, H. (1996). *Geometric Measure Theory*. Springer, Berlin.

Fournier, N. and Guillin, A. (2015). On the rate of convergence in wasserstein distance of the empirical measure. *Probability Theory and Related Fields*, 162(3-4):707–738.

Gilks, W. R. and Berzuini, C. (2001). Following a moving target—monte carlo inference for dynamic bayesian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(1):127–146.

Givens, G. H. and Raftery, A. E. (1996). Local adaptive importance sampling for multivariate densities with strong nonlinear relationships. *Journal of the American Statistical Association*, 91(433):132–141.

Kim, Y., Roh, D., and Lee, M. (2000). Nonparametric adaptive importance samplign for rare event simulation. In *Proceedings of the 2000 Winter Simulation Conference*, pages 767–772.

Lee, Y. T. and Vempala, S. S. (2016). Geodesic walks in polytopes. *arXiv preprint arXiv:1606.04696*.

Liu, J. S. (2008). *Monte Carlo strategies in scientific computing*. Springer Science & Business Media.

Ma, W., Trusina, A., El-Samad, H., Lim, W. A., and Tang, C. (2009). Defining network topologies that can achieve biochemical adaptation. *Cell*, 138(4):760–773.

Michaelis, L., Menten, M., Johnson, K., and Goody, R. (2011). The original michaelis constant: translation of the 1913 michaelis-menten paper. *Biochemistry*, 50(39):8264–8269.

Rhee, C.-H., Zhou, E., and Qiu, P. (2014). An iterative algorithm for sampling from manifolds. In *Proceedings of the 2014 Winter Simulation Conference*, pages 574–585. IEEE Press.

Robert, C. P. (2004). *Monte carlo methods*. Wiley Online Library.

Rosenthal, J. S., Dote, A., Dabiri, K., Tamura, H., and Sheikholeslami, A. (2019). Jump markov chains and rejection-free metropolis algorithms. *arXiv preprint arXiv:1910.13316*.

Santner, T. J., Williams, B. J., and Notz, W. I. (2013). *The design and analysis of computer experiments*. Springer Science & Business Media.

Smith, R. L. (1984). Efficient monte carlo procedures for generating points uniformly distributed over bounded regions. *Operations Research*, 32(6):1296–1308.

Villani, C. (2008). *Optimal transport: old and new*, volume 338. Springer Science & Business Media.

Zhang, P. (1996). Nonparametric importance sampling. *Journal of the American Statistical Association*, 91(435):1245–1253.

Zlochin, M. and Baram, Y. (2002). Efficient nonparametric importance sampling for bayesian learning. In *Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on*, volume 3, pages 2498–2502. IEEE.