



Université catholique de Louvain
Institute of Statistics, Biostatistics and Actuarial Sciences

Enhancing Monte Carlo integration by control variates and statistical learning

Aigerim ZHUMAN

A thesis submitted to the
Université catholique de Louvain
in partial fulfillment of the
requirements for the degree of
Doctor of Sciences

Thesis Committee:

Prof. Johan Segers	UCLouvain	Advisor
Prof. Rainer von Sachs	UCLouvain	Chairman
Prof. Eugen Pircalabelu	UCLouvain	Secretary
Prof. François Portier	ENSAI	
Prof. Philippe Lambert	ULiège	

October 2024

Acknowledgments

First of all, I would like to express my gratitude to my supervisor, Prof. Johan Segers, for his continuous guidance and support. It has been a privilege and pleasure to work with you, Johan. You were always available and responsive when we needed to meet conference or bureaucratic deadlines. You brought more structure and order to my scientific routine. Thank you for organizing lunches and outdoor activities with your other PhD students, postdocs, and collaborators. I appreciate the great opportunity you provided to me.

My special thanks go to Prof. François Portier for being part of my PhD committee from the beginning, along with his inspiring and brilliant ideas and the productive collaboration that led to outstanding scientific papers. You are truly a Monte Carlo mastermind. Additionally, I want to express my gratitude to Dr. Rémi Leluc for his valuable expertise and contributions to our collaborative work.

I would also like to thank Prof. Eugen Pircalabelu, who has been part of my PhD committee from the very beginning. Thank you, Eugen, for the valuable discussions about regularization methods in regression, as well as for your remarks and questions during the confirmation test and private defense. My appreciation also goes to Prof. Rainer von Sachs and Prof. Philippe Lambert for agreeing to be part of my thesis committee. I am grateful for their careful reading of my thesis and for their valuable remarks and questions during the private defense.

Next, I would like to thank my PhD and postdoc colleagues. Thank you, Ensiyeh and Anas, for being nice colleagues and friends, and for the enjoyable conversations during lunch and outside the institute. I also want to thank my officemates Hortense, Benjamin D., and Hugo for the interesting and informative discussions, and for helping me learn French. Additionally, I want to thank Charlotte, Chikêola, Keivan, Mengxue, Patricia, and Vanessa for creating a friendly atmosphere on the third floor. I would also like to thank all the PhD students and postdocs at ISBA, including Alexandre, Antoine, Aurèle, Charles-Guy, Edouard, Fadoua, Fanny, Gabriel, Gilles, Haotian, Hugues, Jean-Loup, Jeongjin, John-John, Lara, Lise, Luc, Madeline, Mathilde, Morine, Nathalie Lu., Oussama, Quentin, Rebecca, Shuang, Sophie Mat., Stefka, and Stéphane.

My special gratitude goes to the administrative team, including Nancy, Sophie Mal., Tatiana, and Nadja, for their assistance with any practical questions. I would also like to thank the SMCS team, including Alain, Aurélie, Benjamin C., Catherine, Céline, Nathalie Le., Lieven, Séverine, and Vincent.

Undoubtedly, this thesis would not have been possible without the support of my family. I am infinitely grateful to my mom, Viktoriya, for her constant support and belief in me throughout my life. I also want to thank my partner, Sam, for being by my side over the past four years, as well as Sam's parents, Véronique and Peter, who have welcomed me into their family.

I gratefully acknowledge research grants from the National Bank of Belgium and of the Research Council of UCLouvain.

Contents

1	Introduction	1
1.1	Naive Monte Carlo	2
1.2	Variance reduction techniques	5
1.2.1	Control variates	6
1.2.2	Importance sampling	7
1.3	Nearest neighbors	9
1.4	Random forests	10
1.5	Outline of the thesis	12
2	Adaptive Importance Sampling and Control Variates	15
2.1	Introduction	15
2.2	Preliminaries on Monte Carlo integration	18
2.3	Combining adaptive importance sampling with control variates	20
2.4	Theoretical properties of the AISCV estimate	22
2.5	Practical considerations	24
2.5.1	Control variate constructions	24
2.5.2	Bayesian inference	26
2.6	Numerical illustration	26
2.6.1	Synthetic examples	27
2.6.2	Real-world examples	28
2.7	Appendix	30
2.7.1	Auxiliary results	30
2.7.2	Additional properties of AISCV estimator	32
2.7.3	Proofs of the main results	34
2.7.4	Additional numerical results	39
3	Control Variates and Nearest Neighbors	47

3.1	Introduction	47
3.2	From nearest neighbors to control neighbors	50
3.2.1	Nearest neighbors	51
3.2.2	Control neighbors	53
3.3	Main results	54
3.3.1	Distributions on metric spaces	54
3.3.2	Moments of nearest-neighbor distance	56
3.3.3	Root mean squared error bounds	57
3.3.4	Concentration inequalities	58
3.3.5	Optimal L_p -approximation with nearest neighbors	59
3.4	Numerical experiments	60
3.4.1	Practical implementation	61
3.4.2	Integration on various spaces: $[0, 1]^d, \mathbb{R}^d, O_m(\mathbb{R})$ and $\mathbb{S}^{q-1}$	63
3.4.3	Monte Carlo integration for optimal transport	66
3.5	Discussion	69
3.6	Appendix	69
3.6.1	Auxiliary results	69
3.6.2	Proofs of Lemmas	70
3.6.3	Proofs of Propositions	73
3.6.4	Proofs of Theorems	76
3.6.5	Additional experiments	90
4	Spherical Harmonics as Control Variates	95
4.1	Introduction	95
4.2	Optimal transport distances: Wasserstein and sliced-Wasserstein	98
4.2.1	Wasserstein distance	98
4.2.2	Sliced-Wasserstein distance	99
4.3	Monte Carlo with control variates	100
4.4	Sliced-Wasserstein estimate with spherical harmonics as control variates	102
4.4.1	Spherical harmonics	102

4.4.2	Spherical harmonics control variates estimate	104
4.5	Numerical experiments	105
4.5.1	Synthetic data: multivariate Gaussians	106
4.5.2	Empirical distributions of 3D point clouds	107
4.5.3	Sliced-Wasserstein and Kernel SVM	108
4.6	Theoretical analysis	108
4.6.1	Elementary properties	109
4.6.2	Asymptotic error bound	110
4.7	Conclusion	111
4.8	Appendix	112
4.8.1	Algorithms for sliced-Wasserstein estimates	112
4.8.2	Numerical details and additional results	115
4.8.3	Mathematical proofs	124
4.8.4	Additional results on spherical harmonics	125
5	Random Forest Control Variates	133
5.1	Introduction	133
5.2	Decision trees and random forests	135
5.2.1	Regression trees	135
5.2.2	Random forests	137
5.3	Control variates	139
5.4	Numerical study	140
5.5	Appendix	144
5.5.1	Mondrian forests	145
5.5.2	Additional numerical simulations	146
6	Conclusion and discussion	149

Introduction

1

Monte Carlo methods have their roots in the 1940s and are closely associated with mathematician Stanisław Ulam. While recovering from an illness, Ulam was playing solitaire and pondering the probability of winning a particular card game. He realized that instead of attempting a combinatorial analysis, he could simulate numerous random games to estimate the probability of winning. This insight laid the groundwork for Monte Carlo methods. Collaborating with John von Neumann and others, Ulam formalized these ideas, and the method was named "Monte Carlo" after the famous casino in Monaco, reflecting the element of chance and randomness inherent in the approach (Eckhardt, 1987; Metropolis, 1987). Monte Carlo methods are mainly used in three distinct problem classes: optimization, numerical integration, and generating draws from a probability distribution. This thesis focuses specifically on the integration problem.

Monte Carlo integration is a powerful numerical technique used to approximate definite integrals. It works by randomly sampling points within the integration domain and averaging function evaluations over these points. Unlike methods that follow predetermined steps, Monte Carlo integration relies on randomness that allows it to estimate integrals in situations where deterministic techniques struggle, particularly in high-dimensional spaces. This method is widely used in fields like physics, finance, and engineering for its flexibility and ability to handle complex integrals efficiently.

The primary objective of this thesis is to enhance the Monte Carlo integration method by means of variance reduction techniques. Specifically, our focus is on the method of control variates, introduced in Section 1.2.1. Additionally, during our research, we explored another variance reduction technique called adaptive importance sampling, as presented in Section 1.2.2, and further examined its combination with control variates. Besides, we introduce the techniques of nearest neighbors in Section 1.3 and random forests in Section 1.4, which we further employ for constructing the control variates.

The motivation for using the method of control variates is that it is particularly well-suited to handle a large number of integrands simultaneously. This is because the quadrature weights and points used in the control-variate enhanced Monte Carlo method are independent of the specific integrand. This

independence implies that these weights and points need to be calculated only once and can then be applied uniformly across all integrands. For example, in Bayesian inference, particularly when dealing with models that involve multiple parameters, control variates are instrumental in computing posterior expectations efficiently. Each parameter in the model can be considered as a component of the vector of integrands. However, when dealing with a whole family of functions to be integrated simultaneously, the challenge lies in uniformly controlling the integration error across all integrands.

Over the past years, numerous research studies have been conducted on the method of control variates. In Portier and Segers (2019), a central limit theorem is established for the integration error when the number of control variates tends to infinity. Both Mira et al. (2013) and Oates et al. (2017) focus on Stein control variates using the derivatives of the log target. While the construction of control variates in Mira et al. (2013) is based on a parametric class of polynomials, Oates et al. (2017) proposes a non-parametric extension of control variates. In Leluc et al. (2021) and South et al. (2023), the authors employ regularization for control variate selection. In Wan et al. (2020), control variates are proposed to be constructed with neural networks.

1.1 Naive Monte Carlo

Let $f \in L_2(\mu)$ be a square integrable, real-valued function on a probability space $(\mathcal{X}, \mathcal{F}, \mu)$ of which we would like to calculate the integral

$$I(f) = \int_{\mathcal{X}} f(x) d\mu(x). \quad (1.1)$$

Also, the integral above is the expectation of the function f with respect to the probability measure μ , i.e., $E_{\mu}[f] = I(f)$.

The naive Monte-Carlo estimator of $I(f)$ based on independent random variables $X_1, \dots, X_n$ taking values in $\mathcal{X}$ and with common distribution μ is

$$I_n(f) = \frac{1}{n} \sum_{i=1}^n f(X_i). \quad (1.2)$$

According to the law of large numbers, this estimator converges to the real value of the integral with probability one, i.e., for any positive number ϵ ,

$$\lim_{n \rightarrow \infty} P(|I_n(f) - I(f)| < \epsilon) = 1.$$

In addition, it is unbiased, which means that $E_{\mu}[I_n(f)] = I(f)$. Moreover, by the central limit theorem, the Monte Carlo integration error $I_n(f) - I(f)$ is asymptotically $\mathcal{N}(0, \sigma^2(f)n^{-1})$ distributed, where $\sigma^2(f) = E_{\mu}[(f - E_{\mu}[f])^2]$.

Example 1.1 (Molenberghs and Verbeke (2000)). Consider a possible application of the Monte Carlo integration method for inference on linear mixed models. These models are extensively used in longitudinal studies to analyze repeated measurement data, such as monitoring the progression of diseases over time or assessing the effectiveness of treatments.

A linear mixed model is given by

$$Y_i = X_i\beta + Z_ib_i + \varepsilon_i,$$

where Y_i is the n_i -dimensional response vector for subject i , $1 \leq i \leq N$, N is the number of subjects, X_i and Z_i are $(n_i \times q)$ and $(n_i \times d)$ dimensional matrices of known covariates, β is a q -dimensional vector containing the fixed effects, b_i is the d -dimensional vector containing the random effects with zero mean and $(d \times d)$ covariance matrix D , and ε_i is an n_i -dimensional vector of error components with zero mean and $(n_i \times n_i)$ covariance matrix Σ_i . For instance, if $b_i \sim \mathcal{N}(0, D)$ and $\varepsilon_i \sim \mathcal{N}(0, \Sigma_i)$, and b_i and ε_i are independent, then $Y_i | b_i \sim \mathcal{N}(X_i\beta + Z_ib_i, \Sigma_i)$.

The integration problem inherent to this model is to obtain the marginal density function of Y_i , which is given by

$$p(y_i) = \int p(y_i | b_i)p(b_i)db_i,$$

where $p(y_i | b_i)$ and $p(b_i)$ are the density functions of $y_i | b_i$ and b_i . The integral above is an important component in the construction of the likelihood function, which in turn is necessary for estimating unknown coefficients β .

Example 1.2 (Glasserman (2004)). Another possible application of the Monte Carlo integration method is the calculation of the expected present value of the payoff of a European call option on a stock. This type of option gives the holder the right to buy an underlying asset, e.g., a stock, at a specified strike price K on a specified expiration date T . Let $S(t)$ denote the price of the stock at time t . The payoff of the call option at time T is

$$(S(T) - K)^+ = \max\{0, S(T) - K\}.$$

To get the present value of this payoff we multiply by a discount factor e^{-rT} , with r a continuously compounded interest rate. We denote the expected present value by $E[e^{-rT}(S(T) - K)^+]$, which is the integral of interest.

The evolution of the stock price is described by the Black–Scholes model. According to this model, the terminal stock price is represented as

$$S(T) = S(0) \exp\left((r - \sigma^2/2)T + \sigma\sqrt{T}Z\right),$$

where σ is the volatility of the stock and Z is a standard normal random variable.

For a European call option, the payoff is relatively straightforward, depending solely on the underlying asset's price at maturity. However, when pricing exotic options, such as Asian or barrier options, using naive Monte Carlo, computational challenges arise due to their complex payoff structures and path dependency. For instance, the payoff of an Asian option depends on the average of the underlying asset's price over a certain period, rather than just its price at maturity. This requires simulating the entire path of the underlying asset over time, not just its terminal value, which increases the computational load. Additionally, the payoff distribution of Asian options can exhibit higher variance due to the averaging process, causing the standard Monte Carlo method to converge more slowly to the true option price.

Example 1.3 (Kolouri et al. (2019)). The naive Monte Carlo method can also be applied in the domain of optimal transport, which focuses on measuring the distance between probability distributions. Let, for $\mathcal{Y} \subseteq \mathbb{R}^d$, $\mathcal{P}(\mathcal{Y})$ denote the set of probability measures supported on $\mathcal{Y}$ and let $p \in [1, \infty)$. The Wasserstein distance of order p between $P, Q \in \mathcal{P}(\mathcal{Y})$ is

$$W_p^p(P, Q) = \inf_{\pi \in \Pi(P, Q)} \int_{\mathcal{Y} \times \mathcal{Y}} \|x - y\|^p d\pi(x, y),$$

where $\Pi(P, Q) \subset \mathcal{P}(\mathcal{Y} \times \mathcal{Y})$ denotes the set of couplings for (P, Q) , i.e., probability measures whose marginals with respect to the first and second variables are P and Q respectively. This distance suffers from a high computational cost. However, the Sliced-Wasserstein distance overcomes this issue by introducing the concept of slicing the probability distributions onto a hypersphere, enabling more computationally efficient calculations.

Let $\mathbb{S}^{d-1} = \{\theta \in \mathbb{R}^d : \|\theta\| = 1\}$ be the unit sphere in $\mathbb{R}^d$ and for $\theta \in \mathbb{S}^{d-1}$ let $\theta^\star : \mathbb{R}^d \rightarrow \mathbb{R}$ denote the linear map $\theta^\star(x) = \langle \theta, x \rangle$ for $x \in \mathbb{R}^d$. Let μ denote the uniform distribution on $\mathbb{S}^{d-1}$, that is, the normalized volume measure. The Sliced-Wasserstein (SW) distance of order p based on μ is defined for $P, Q \in \mathcal{P}(\mathcal{Y})$ as

$$SW_p^p(P, Q) = \mathbb{E}_\theta[W_p^p(\theta_\#^\star P, \theta_\#^\star Q)] = \int_{\mathbb{S}^{d-1}} W_p^p(\theta_\#^\star P, \theta_\#^\star Q) d\mu(\theta), \quad (1.3)$$

where $g_\# \xi$ is the *push-forward* measure of $\xi \in \mathcal{P}(\mathbb{R}^d)$ by a measurable function g on $\mathbb{R}^d$. For $\theta \in \mathbb{S}^{d-1}$, the measures $\theta_\#^\star P$ and $\theta_\#^\star Q$ are the distributions of the projections $\langle \theta, X \rangle$ and $\langle \theta, Y \rangle$ of random vectors X and Y with distributions P and Q , respectively. The integral in Eq. (1.3) is the integral of interest.

The SW distance finds an application in image classification problems, particularly when used in combination with Kernel SVM. Support Vector Machine (SVM) is a powerful supervised learning algorithm used for classification tasks. It works by finding the optimal hyperplane that separates data points of different classes with the maximum margin. However, in many cases, the data is not linearly separable in the original feature space. This is where Kernel SVM becomes necessary.

A kernel function allows SVM to operate in a transformed feature space, where the data might become linearly separable. The kernel function implicitly maps the input data into a higher-dimensional space without requiring explicit computation of this mapping, enabling the SVM to find a suitable hyperplane even in complex, non-linear cases.

The Sliced Wasserstein Kernel (Kolouri et al., 2016) is a good choice for image classification because it effectively captures the detailed structures within images, which traditional linear or polynomial kernels might miss. By using the Sliced Wasserstein distance, this kernel compares images in a way that matches their natural structure, making it especially good for handling complex image data.

However, a challenge arises because computing the SW distance using the standard Monte Carlo method can be computationally intensive, especially when dealing with large datasets or high-dimensional images. This computational burden is due to the need for multiple projections and pairwise distance calculations, which can significantly increase processing time and resource requirements.

1.2 Variance reduction techniques

Despite the fact that Monte Carlo integration offers an effective method for estimating complex integrals using random sampling, it may face difficulties when dealing with functions that have high variance or it may exhibit slow convergence. In such cases, standard Monte Carlo integration may require a large number of samples to achieve acceptable accuracy, which can be computationally expensive and time-consuming. Variance reduction techniques are essential in these situations to improve the efficiency and precision of Monte Carlo integration. They enable more accurate integral approximations to be obtained with fewer samples. This not only speeds up the computation but also enhances the reliability of the results.

1.2.1 Control variates

The concept behind control variates involves utilizing auxiliary functions $\varphi_1, \dots, \varphi_m$, for which the integrals $I(\varphi_j) = \int_{\mathcal{X}} \varphi_j(x) d\mu(x)$, $j = 1, \dots, m$, are already known. Without loss of generality, assume $I(\varphi_j) = \int_{\mathcal{X}} \varphi_j(x) d\mu(x) = 0$ for all $j = 1, \dots, m$. For every column vector $\beta \in \mathbb{R}^m$, we have $I(f) = I(f - \beta^\top \varphi)$, where $\varphi = (\varphi_1, \dots, \varphi_m)^\top$. Then we obtain an unbiased estimator of $I(f)$ as follows

$$I_n^{(\text{cv})}(f, \beta) = \frac{1}{n} \sum_{i=1}^n [f(X_i) - \beta^\top \varphi(X_i)]. \quad (1.4)$$

The variance of the estimator above is equal to $n^{-1} \sigma^2(f - \beta^\top \varphi)$. By minimizing it over β , we obtain an oracle coefficient vector

$$\beta_\star = I(\varphi \varphi^\top)^{-1} I(\varphi f). \quad (1.5)$$

Hence, the minimal asymptotic variance is

$$\sigma^2(f - \beta_\star^\top \varphi) = \sigma^2(f) - I(f \varphi^\top) I(\varphi \varphi^\top)^{-1} I(\varphi f) \leq \sigma^2(f). \quad (1.6)$$

In practice, $\beta_\star$ in (1.5) is unknown and needs to be estimated. Any estimator $\hat{\beta}$ of $\beta_\star$ produces a control variate estimator

$$I_n^{(\text{cv})}(f) = I_n^{(\text{cv})}(f, \hat{\beta}) = \frac{1}{n} \sum_{i=1}^n [f(X_i) - \hat{\beta}^\top \varphi(X_i)]. \quad (1.7)$$

As soon as $\hat{\beta} \xrightarrow{d} \beta_\star$, we have according to Theorem 1 in Glynn and Szechtman (2002) the minimal asymptotic variance in the limit,

$$\sqrt{n} \left(I_n^{(\text{cv})}(f) - I(f) \right) \xrightarrow{d} \mathcal{N}(0, \sigma^2(f - \beta_\star^\top \varphi)), \quad n \rightarrow \infty. \quad (1.8)$$

Ordinary least squares estimator. There exist multiple options to estimate $\beta_\star = I(\varphi \varphi^\top)^{-1} I(\varphi f)$ (Glynn and Szechtman, 2002). The more common estimator is

$$\hat{\beta}_n^{\text{OLS}} = G_n^{-1} \{I_n(f \varphi) - I_n(f) I_n(\varphi)\}, \quad (1.9)$$

where $G_n = I_n(\varphi \varphi^\top) - I_n(\varphi) I_n(\varphi^\top)$ is the empirical covariance matrix of the control variates. The resulting ordinary least squares Monte Carlo estimator is

$$\hat{\alpha}_n^{\text{OLS}} = I_n^{(\text{cv})}(f, \hat{\beta}_n^{\text{OLS}}) = I_n(f) - \{I_n(f \varphi^\top) - I_n(f) I_n(\varphi^\top)\} G_n^{-1} I_n(\varphi). \quad (1.10)$$

The terminology originates from the well-known property (Glasserman, 2004) that

$$(\hat{\alpha}_n^{\text{OLS}}, \hat{\beta}_n^{\text{OLS}}) = \arg \min_{a \in \mathbb{R}, b \in \mathbb{R}^m} \sum_{i=1}^n (f(X_i) - a - b^\top \varphi(X_i))^2, \quad (1.11)$$

The identity (1.11) arises from the normal equations in the multiple linear regression model

$$f(X_i) = \alpha + \beta_\star^\top \varphi(X_i) + \varepsilon_i, \quad i = 1, \dots, n, \quad (1.12)$$

with dependent variable $f(X_i)$, explanatory variables $\varphi_1(X_i), \dots, \varphi_m(X_i)$ and errors ε_i . The variance of the errors is equal to the asymptotic variance of the ordinary least squares Monte Carlo estimator $I(\varepsilon^2) = \sigma^2(f - \beta_\star^\top \varphi)$.

Example 1.4 (Portier and Segers (2019)). Consider the integration problem on $[-1, 1]$ with respect to the uniform distribution μ . Let f be $k+1$ times continuously differentiable for some integer $k \geq 1$. In this case, a natural choice of control variates is the Legendre polynomials, as they are orthogonal on $[-1, 1]$ with respect to the uniform distribution. Specifically, let $\varphi_j = L_j$, where L_j is the Legendre polynomial of degree j for $j = 1, \dots, m_n$. We consider the set-up where the number of control variates m_n grows with n , but not too fast. In this specific situation, the condition $m_n = o(n^{1/3})$ must be satisfied. Then, the convergence rate of $\hat{\alpha}_n^{\text{OLS}}$ is $O(m_n^{-k-1/2} n^{-1/2})$, which is faster than the rate of the naive Monte Carlo method, with the acceleration depending on the smoothness of the integrand.

1.2.2 Importance sampling

Importance sampling refers to a collection of Monte Carlo methods where a mathematical expectation with respect to a target distribution is approximated by a weighted average of random draws from another distribution, particularly useful when direct sampling from the target distribution is difficult, expensive, or computationally infeasible. Let $p(x)$ be a probability density on $\mathcal{X} \subseteq \mathbb{R}^d$ and suppose we wish to compute an expectation $I(f) = \mathbb{E}_p[f]$, with

$$I(f) = \int_{\mathcal{X}} f(x) p(x) dx.$$

Then for any probability density $q(x)$, s.t. $q(x) > 0$ whenever $f(x)p(x) \neq 0$, one has

$$I(f) = \mathbb{E}_q[wf] = \int_{\mathcal{X}} w(x) f(x) q(x) dx, \quad w(x) = \frac{p(x)}{q(x)}.$$

Therefore, a sample of independent draws $X_1, \dots, X_n$ from $q(x)$ can be used to estimate $I(f)$ by

$$I_n^{(\text{is})}(f) = \frac{1}{n} \sum_{i=1}^n w_i f(X_i), \quad w_i = \frac{p(X_i)}{q(X_i)}. \quad (1.13)$$

From Theorem 9.1 in Owen (2013), $E_q(I_n^{(\text{is})}(f)) = I(f)$ and $\text{Var}_q(I_n^{(\text{is})}(f)) = \sigma_q^2/n$ with

$$\sigma_q^2 = \int_{\mathcal{X}} \frac{(f(x)p(x))^2}{q(x)} dx - I(f)^2 = \int_{\mathcal{X}} \frac{(f(x)p(x) - I(f)q(x))^2}{q(x)} dx.$$

In many applications the density $p(x)$ is known only up to a normalizing constant. In this case one replaces (1.13) with the normalized estimator

$$I_{\text{norm}}^{(\text{is})}(f) = \frac{\sum_{i=1}^n w_i f(X_i)}{\sum_{i=1}^n w_i}. \quad (1.14)$$

Theorem 9.2 in Owen (2013) shows that $P\left(\lim_{n \rightarrow \infty} I_{\text{norm}}^{(\text{is})}(f) = I(f)\right) = 1$.

Example 1.5 (Owen (2013)). Importance sampling can be used to estimate the probability of a rare event. Let's consider computing the probability that a random walk with negative drift and starting at zero reaches the level $b > 0$ before reaching the level $a \leq 0$. Suppose X_j are independent and identically distributed random variables with density p having negative mean and satisfying $P(X > 0) > 0$. Let $S_0 = 0$, and for $m \geq 1$, let $S_m = \sum_{j=1}^m X_j$. The process S_m is a random walk starting at 0, taking p -distributed jumps, and M represents the first time, after 0, that the process is outside the interval $(a, b]$. The desired quantity $P(S_M > b)$ gives the probability that at the time of the first exit from $(a, b]$, the random walk is positive. However, the walk has a negative drift so that if b is large enough, then $S_M > b$ is a rare event.

In this specific context, we can interpret X_j as a gambler's winnings, with losses represented as negative winnings, at the j -th play of the game. The game continues until either the gambler's winnings reach b , or the losses reach $|a|$. Then $P(S_M > b)$ represents the probability that the game concludes with the gambler having positive winnings.

Consider a normal random walk. Suppose that $X_j \sim p$, the $\mathcal{N}(\theta_0, \sigma^2)$ density, with $\theta_0 < 0$. Let q be the $\mathcal{N}(\theta_\star, \sigma^2)$ density. For simplicity, assume $\sigma = 1$. Then

$$\frac{p(x)}{q(x)} = \exp((\theta_0 - \theta_\star)x).$$

The key idea is to reverse the drift to make the walk tend to move upward towards b , which makes the rare event more likely to happen in the simulations. To do this, we set $\theta_\star = -\theta_0$. This means that if the original drift θ_0 is

negative, the new drift $\theta_\star$ will be positive, thus reversing the direction of the random walk's tendency. Then the importance sampling estimator for the probability of the rare event $P(S_M > b)$ is

$$\hat{\mu}_q = \frac{1}{n} \sum_{i=1}^n \mathbb{1}_{\{S_{M,i} > b\}} \exp(2\theta_0 S_{M,i}).$$

Adaptive importance sampling (Portier and Delyon, 2018). The choice of q can be a delicate matter. A poor choice could yield an infinite variance for the estimator of $I(f)$. Adaptive importance sampling helps to overcome this issue by enabling the finding of a better importance distribution q . In this approach, we consider a family of tractable densities $(q_i)_{i \geq 0}$ that evolve towards a density q_{opt} which optimizes some criterion. The initial density q_0 is fixed, but for $i \geq 1$, the density q_i is determined based on the previously sampled particles $X_1, \dots, X_i$. Thus, after sampling $X_1, \dots, X_i$, the next particle X_{i+1} is drawn from q_i . Therefore, similarly to (1.14), the normalized adaptive importance sampling estimator of $I(f)$ is defined as

$$I_{\text{norm}}^{(\text{ais})}(f) = \frac{\sum_{i=1}^n w_i f(X_i)}{\sum_{i=1}^n w_i} \quad \text{where} \quad w_i = \frac{p(X_i)}{q_{i-1}(X_i)} \quad \text{for } i = 1, \dots, n. \quad (1.15)$$

1.3 Nearest neighbors

K-nearest neighbors (Biau and Devroye, 2015; Hastie et al., 2001) is a non-parametric method used for estimating the value of a target variable based on the values of its neighboring data points. It is widely used in various fields, including machine learning, pattern recognition, and data mining, due to its simplicity and effectiveness. We are particularly interested in the case when $k = 1$, known as the nearest neighbor method. In this scenario, the prediction for a data point is simply the value of its closest neighbor in the feature space.

Let $X_1, \dots, X_n$ be an independent and identically distributed random sample in $\mathbb{R}^d$, drawn from a distribution μ . Let $\|\cdot\|$ denote the Euclidean norm and let $x \in \mathbb{R}^d$ be a generic point.

The nearest neighbor $\hat{N}_n(x)$ of x among $X_1, \dots, X_n$ and the associated distance $\hat{\tau}_n(x)$ are

$$\hat{N}_n(x) \in \arg \min_{Y \in \{X_1, \dots, X_n\}} \|x - Y\|, \quad \hat{\tau}_n(x) = \|\hat{N}_n(x) - x\|. \quad (1.16)$$

When the arg min is not unique, $\hat{N}_n(x)$ is defined as the one point among the arg min having the smallest index.

In the same way we define the leave-one-out neighbor. Let $i \in \{1, \dots, n\}$ and $\mathcal{X}_n^{(i)} = \{X_1, \dots, X_n\} \setminus \{X_i\}$. The leave-one-out neighbor of $x \in \mathbb{R}^d$ and the associated distance $\hat{\tau}_n^{(i)}(x)$ are

$$\hat{N}_n^{(i)}(x) \in \arg \min_{Y \in \mathcal{X}_n^{(i)}} \|x - Y\|, \quad \hat{\tau}_n^{(i)}(x) = \|\hat{N}_n^{(i)}(x) - x\|. \quad (1.17)$$

Figure 1.1 illustrates the nearest neighbor and the leave-one-out neighbor for the point x within the set of points $X_1, \dots, X_7$.

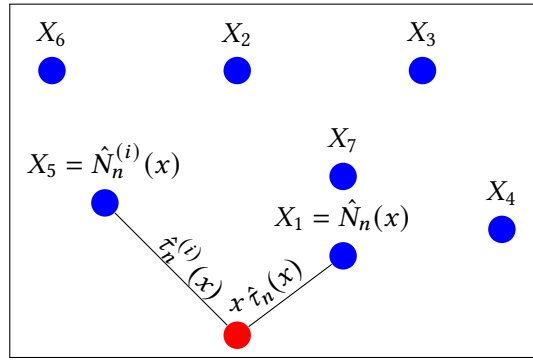


Figure 1.1: Illustration of leave-one-out/nearest neighbor of x , where X_1 is the nearest neighbor of x and X_5 is the leave-one-out neighbor of x , specifically for $i = 1$.

Based on Eq. (1.16) and (1.17), the nearest neighbor and leave-one-out neighbor estimators of a real-valued function f defined on the support of μ are expressed as

$$\hat{f}_n^{(\text{NN})}(x) = f(\hat{N}_n(x)) \quad (1.18)$$

and

$$\hat{f}_{n,i}^{(\text{loo})}(x) = f(\hat{N}_n^{(i)}(x)) \quad (1.19)$$

respectively.

1.4 Random forests

Random forests (Louppe, 2014) are a versatile and powerful ensemble learning method employed for both classification and regression tasks in machine learning. This method involves constructing numerous decision trees during training and generating predictions by either selecting the mode of the classes (for classification) or averaging the predictions (for regression) of the individual trees, as illustrated in Figure 1.2. In our case, we are focused on function estimation and are therefore primarily interested in the latter application.

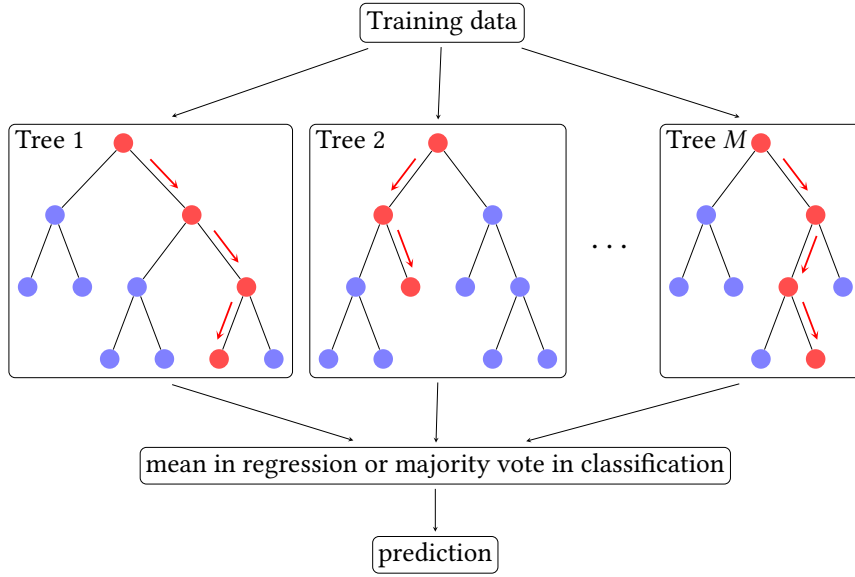


Figure 1.2: Diagram of a random forest algorithm with M trees

Without loss of generality, consider one regression tree (Breiman et al., 1984). The idea behind this method is to partition the input space into a set of regions based on the feature values. Each region corresponds to a leaf node of the tree, where a simple model, typically the mean of the target variable for the training instances within that region, is used to make predictions. A leaf node is a terminal node in the tree that does not split further and contains the output value for its region. The tree is built by recursively splitting the data at each node according to a feature and a threshold that maximize a certain criterion, such as the reduction in mean squared error. This process continues until a stopping condition is met, such as a maximum depth or a minimum number of samples per leaf. By capturing complex patterns and interactions between features, a single regression tree can provide a piecewise constant approximation of the target function.

Let $X_1, \dots, X_n$ be an independent and identically distributed random sample in $\mathbb{R}^d$, drawn from a distribution μ . These values are divided into J distinct, non-overlapping regions $R_1, R_2, \dots, R_J$ as explained above. Let f be a real-valued function defined on the support of μ . For any observation falling into region R_j , the prediction is calculated as the mean of the response values $y_i = f(X_i)$ for all n_j observations within that region R_j , i.e.,

$$\hat{y}_j = \frac{1}{n_j} \sum_{X_i \in R_j} y_i.$$

Therefore, the regression tree estimator of f is

$$\hat{f}^{(\text{rt})}(x) = \sum_{j=1}^J \hat{y}_j \mathbb{1}_{R_j}(x).$$

To obtain the random forest estimator with M trees, we consider an ensemble of such regression trees. Let $\hat{f}_m^{(\text{rt})}(x)$ denote the m -th regression tree estimator. The random forest estimator is the average of the predictions from all M trees:

$$\hat{f}^{(\text{rf})}(x) = \frac{1}{M} \sum_{m=1}^M \hat{f}_m^{(\text{rt})}(x). \quad (1.20)$$

In ensembles of trees, we introduce randomness to ensure that all trees differ from each other. This can be achieved through various methods, such as using bootstrap samples to build each tree, considering random subsets of features, or choosing random thresholds for splitting at each node. This diversity among the trees enhances the robustness and accuracy of the ensemble estimator.

1.5 Outline of the thesis

Chapter 2. This chapter is devoted to combining two powerful variance reduction techniques: adaptive importance sampling and control variates, both explained in Section 1.2. As a result, we obtain the following adaptive importance sampling estimate with control variates (AISCV):

$$\begin{aligned} \mathbf{I}_n^{(\text{aiscv})}(f) &= \mathbf{I}_n^{(\text{aiscv})}(f, \hat{\beta}_n) = \hat{\alpha}_n \quad \text{where} \\ (\hat{\alpha}_n, \hat{\beta}_n) &\in \arg \min_{(a, b) \in \mathbb{R} \times \mathbb{R}^m} \sum_{i=1}^n w_i (f(X_i) - a - b^\top \varphi(X_i))^2, \end{aligned} \quad (1.21)$$

where the importance weights w_i are defined as in Eq (1.15). The estimate is well-defined only if the minimizer $\hat{\alpha}_n$ is unique—the minimizer $\hat{\beta}_n$ need not be. Moreover, we establish a quadrature rule where the quadrature points and weights are independent of the integrand, offering a computational advantage when dealing with multiple integrands. In addition, we demonstrate several special properties of the proposed estimate, such as:

- Exact integration when $f = \alpha + \beta^\top \varphi$ for $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}^m$.
- Invariance of the estimate based on the control variates $A\varphi$ for any invertible matrix $A \in \mathbb{R}^{m \times m}$.

As the main theoretical result, we derived a non-asymptotic error bound for the AISCV estimator under certain assumptions. Furthermore, our theoretical findings are supported by numerical studies including the following cases: integration on $[0, 1]^d$ with respect to the uniform distribution, integration with respect to a mixture of Gaussian distributions, and several real-world examples in the context of Bayesian inference. The content of this chapter is based on joint work with R. Leluc, F. Portier and J. Segers and is published in the *Advances in Neural Information Processing Systems*.

Chapter 3. In this chapter, we employ the nearest neighbor approach introduced in Section 1.3 to construct the control variates. More specifically, we construct the control variate estimate in the following form:

$$I_n^{(\text{cv})}(f) = \frac{1}{n} \sum_{i=1}^n \left[f(X_i) - \left\{ \hat{f}_n(X_i) - I(\hat{f}_n) \right\} \right], \quad (1.22)$$

where the integrand f belongs to the class of Hölder functions with regularity $s \in (0, 1]$. As a result, we obtain two estimates:

$$\begin{aligned} I_n^{(\text{NN})}(f) &= \frac{1}{n} \sum_{i=1}^n \left[f(X_i) - \left\{ \hat{f}_{n,i}^{\text{loo}}(X_i) - I(\hat{f}_n^{\text{NN}}) \right\} \right], \\ I_n^{(\text{NN-loo})}(f) &= \frac{1}{n} \sum_{i=1}^n \left[f(X_i) - \left\{ \hat{f}_{n,i}^{\text{loo}}(X_i) - I(\hat{f}_{n,i}^{\text{loo}}) \right\} \right], \end{aligned}$$

where $\hat{f}_n^{\text{NN}}$ is the nearest neighbor estimate as in Eq. (1.18) and $\hat{f}_{n,i}^{\text{loo}}$ is the leave-one-out neighbor estimate as in Eq. (1.19). The estimate $I_n^{(\text{NN})}(f)$ is biased but more computationally efficient. Moreover, we show that the root mean squared distance between these estimates is of the order $\mathcal{O}(n^{-1/2-s/d})$ as $n \rightarrow \infty$. For both estimates, we establish asymptotic and non-asymptotic error bounds. The simulation studies involve some examples concerning integration problems on different spaces such as the unit cube $[0, 1]^d$, the Euclidean space $\mathbb{R}^d$, the orthogonal group $O_m(\mathbb{R})$ and the unit sphere $\mathbb{S}^2$. In addition, the proposed method is applied for calculating the Sliced-Wasserstein distance described in Example 1.3 in Section 1.1. The content of this chapter is based on joint work with R. Leluc, F. Portier and J. Segers and is in press in the *Bernoulli Journal*.

Chapter 4. This chapter focuses on constructing control variates for the application discussed in the previous chapter: the calculation of the Sliced-Wasserstein distance. Specifically, since the integration problem is defined on the unit sphere, we propose constructing the control variates using spherical harmonics (Atkinson and Han, 2012; Dai, 2013). Spherical harmonics are

spherical polynomials that form an orthonormal basis for the space of square-integrable functions on the sphere. For instance, on the unit circle, the spherical harmonics are called circular harmonics and correspond to the Fourier basis functions. We introduce a new estimate called the Spherical Harmonics Control Variates (SHCV) estimate. This estimate arises as the least squares estimate for the intercept in a multiple linear regression model, where spherical harmonics are used as explanatory variables, as described in Section 1.2.1. As the main theoretical result, we establish the convergence rate in probability for the SHCV integration error. In addition, numerical experiments are conducted using synthetic data for continuous measures and point clouds for discrete measures. Moreover, the proposed estimate is applied to the Kernel SVM with a Sliced-Wasserstein kernel (Kolouri et al., 2016) for the image classification problem. The content of this chapter is based on joint work with R. Leluc, A. Dieuleveut, F. Portier and J. Segers and is published in the *Proceedings of the 41st International Conference on Machine Learning*.

Chapter 5. In this chapter, we build the control variate estimate in the same way as in Eq. (1.22), but here $\hat{f}_n(x)$ corresponds to the random forest estimate presented in Section 1.4. We focus on several ensemble methods for trees, including Random Forests (Breiman, 2001), Bagging (Breiman, 1996), and Extremely Randomized Trees (Geurts et al., 2006). In both Bagging and Random Forests, trees are constructed based on bootstrap samples. However, Random Forests additionally involve feature selection. In contrast, Extremely Randomized Trees use the entire learning set but introduce randomization by randomly selecting the splitting thresholds for each feature. As a result, we obtain three working estimates. To construct the optimal tree, we conduct numerical experiments using synthetic examples to illustrate how tuning parameters, such as maximal tree depth and the number of trees, affect the accuracy of our proposed estimates. Additionally, these estimates are used to calculate the variance of the asymptotic distribution of U-statistics.

Chapter 6. This chapter presents the overall conclusions drawn from the preceding chapters and suggests possible lines of future research.

Adaptive Importance Sampling and Control Variates | 2

Driven by several successful applications such as in stochastic gradient descent or in Bayesian computation, control variates have become a major tool for Monte Carlo integration. However, standard methods do not allow the distribution of the particles to evolve during the algorithm, as is the case in sequential simulation methods. Within the standard adaptive importance sampling framework, a simple weighted least squares approach is proposed to improve the procedure with control variates. The procedure takes the form of a quadrature rule with adapted quadrature weights to reflect the information brought in by the control variates. The quadrature points and weights do not depend on the integrand, a computational advantage in case of multiple integrands. Moreover, the target density needs to be known only up to a multiplicative constant. Our main result is a non-asymptotic bound on the probabilistic error of the procedure. The bound proves that for improving the estimate's accuracy, the benefits from adaptive importance sampling and control variates can be combined. The good behavior of the method is illustrated empirically on synthetic examples and real-world data for Bayesian linear regression. The content of this chapter is based on the paper

R. Leluc, F. Portier, J. Segers and A. Zhuman. (2022). A Quadrature Rule combining Control Variates and Adaptive Importance Sampling. *Advances in Neural Information Processing Systems*, volume 35, pages 11842–11853.

2.1 Introduction

In recent years, sequential simulation has emerged as a leading approach to compute multidimensional integrals. A key object in sequential simulation is the sequence of distributions, called the policy, from which to generate the random variables, called particles, used to approximate the integrals of interest. The policy is designed to evolve in the course of the algorithm to mimic the target density, which may itself be known only up to a proportionality constant. While the design of algorithms with adaptive policies has been

of major interest recently, only a few studies have focused on using control variates to reduce the variance. This paper provides a new method to incorporate control variates within standard sequential algorithms. The proposed approach significantly improves the accuracy of the initial algorithm, both theoretically and in practice.

The sequential framework. Consider the problem of approximating the integral $\int f p \, d\lambda = \int_{\mathbb{R}^d} f(x) p(x) \, dx$, where λ is the d -dimensional Lebesgue measure, p is a probability density on $\mathbb{R}^d$ and the integrand f is a real-valued function on $\mathbb{R}^d$. For instance, one may think of p as the posterior density in Bayesian inference. Let $(q_i)_{i \geq 0}$ be the policy of the algorithm, i.e., a sequence of probability densities which evolves adaptively depending on previous outcomes. The particles $(X_i)_{i \geq 1}$ are generated sequentially—at iteration i , particle X_i is drawn from q_{i-1} . The integral $\int f p \, d\lambda$ is estimated by the normalized sum $(\sum_{i=1}^n w_i f(X_i)) / (\sum_{i=1}^n w_i)$, where $w_i = p(X_i) / q_{i-1}(X_i)$ are the importance weights. The normalization $\sum_{i=1}^n w_i$ allows to deal with situations where the target density p is known only up to a proportionality constant.

Such an algorithm is part of the *adaptive importance sampling* (AIS) framework. Many different ways have been investigated to update the densities q_i adaptively. Early works that inspired such sequential schemes include Geweke (1989); Kloek and Van Dijk (1978); Oh and Berger (1992) where the sampling policy is chosen out of a parametric family. The parametric approach has been further extended by the Population Monte Carlo framework (Cappé et al., 2008, 2004; Martino et al., 2017). Various asymptotic results have been obtained in Chopin (2004); Douc and Moulines (2008); Portier and Delyon (2018). In Dai et al. (2016); Delyon and Portier (2021); Korba and Portier (2022); Zhang (1996), *nonparametric importance sampling* based on kernel smoothing is studied. The latter bears resemblance to *sequential Monte Carlo* methods (Del Moral et al., 2006; Chopin, 2004), in which the target distribution p changes in the course of the algorithm.

Control variates. Let $\varphi = (\varphi_1, \dots, \varphi_m)^\top$ be a vector of real-valued functions on $\mathbb{R}^d$ such that for each $k = 1, \dots, m$, the integral $\int \varphi_k p \, d\lambda$ is known. Without loss of generality, suppose that $\int \varphi p \, d\lambda = 0$. The functions φ_k are called control variates and can be obtained in different ways. In Bayesian statistics, Stein control variates (Oates et al., 2017) are constructed by applying the second-order Stein operator to functions satisfying certain regularity conditions (Mira et al., 2013). Other control variates might be created by reweighting a function φ^* that satisfies $\int \varphi^* \, d\lambda = 0$ via $\varphi = \varphi^* / p$. The use of control variates is a well studied variance-reduction technique (Glynn and Szechtman, 2002; Owen and Zhou, 2000). The benefits can be established theoretically in terms of error bounds (Oates et al., 2017; Leluc et al., 2021), weak convergence (Portier and Segers, 2019), the excess risk (Belomestny et al.,

2022) and even uniform error bounds over large classes of integrands (Plassier et al., 2020). In practice, the control variates framework has led to efficient procedures in reinforcement learning (Jie and Abbeel, 2010; Liu et al., 2017) and optimization (Wang et al., 2013), to name a few. Importance sampling and control variates in case of a Gaussian target density is explored in Jourdain (2009). The procedure in Kawai (2020) incorporates control variates and is said to involve adaptive importance sampling, but in fact the particles are always sampled from the uniform distribution on the unit cube. To the best of our knowledge, the existing control variate methods do not account for sequential changes in the particle distribution as is the case in AIS.

AISCV estimate. The proposed approach to use control variates within the sequential AIS framework relies on the ordinary least squares expression of control variates (see for instance Portier and Segers (2019)). To account for the policy changes, some re-weighting must be applied. The AISCV estimate of the integral $\int f p \, d\lambda$ is defined as the first coordinate of the solution to the weighted least squares problem

$$(\hat{\alpha}_n, \hat{\beta}_n) = \arg \min_{a \in \mathbb{R}, b \in \mathbb{R}^m} \sum_{i=1}^n w_i (f(X_i) - a - b^\top \varphi(X_i))^2,$$

with w_i the importance weights from before. The AISCV estimate $\hat{\alpha}_n$ has several interesting properties: (a) whenever f is of the form $\alpha + \beta^\top \varphi$ for some $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}^m$, the error is zero, i.e., $\hat{\alpha}_n = \alpha = \int f p \, d\lambda$; (b) the estimate takes the form of a quadrature rule $\hat{\alpha}_n = \sum_{i=1}^n v_{n,i} f(X_i)$, for quadrature weights $v_{n,i}$ that do not depend on the function f and that can be computed by a single weighted least squares procedure; and (c) it can be computed even when p is known only up to a multiplicative constant. Point (a) suggests that when the linear combinations of the functions φ_k span a rich function class, the integration error is likely to be small. Point (b) implies that multiple integrals can be computed just as easily as a single one. Point (c) shows that the approach is applicable for Bayesian computations. In addition, the control variates can be brought into play in a *post-hoc* scheme, after generation of the particles and importance weights, and this for any AIS algorithm.

Main result. The main theoretical result of the paper is a probabilistic, non-asymptotic bound on $\hat{\alpha}_n - \alpha$. Under appropriate conditions, the bound scales as $\tau/\sqrt{n}$, where τ^2 is the scale constant in a sub-Gaussian tail condition on the error variable $\varepsilon = f - \alpha - \beta^\top \varphi$ for $(\alpha, \beta) = \arg \min_{a,b} \int (f - a - b^\top \varphi)^2 p \, d\lambda$. Note that ε has the smallest possible variance one could get using control variates φ . As a consequence, when the space of control variates is well suited for approximating f , the AISCV estimate will be highly accurate. Also, our bound depends only on the linear function space spanned by the control variates $\varphi_1, \dots, \varphi_m$, not on the particular basis chosen in that space. The results rely

on martingale theory, in particular on a concentration inequality for norm-subGaussian martingales in Jin et al. (2019). In the course of the proof, we develop a novel bound on the smallest eigenvalue of certain random matrices, extending an inequality from Tropp (2015) to the martingale case.

Outline. Section 2.2 introduces the general framework of adaptive importance sampling and control variates. Next, Section 2.3 presents the AISCV estimate and the associated quadrature rule. Section 2.4 contains the statements of the theoretical results while Section 2.5 gathers practical considerations, including the construction of control variates. Numerical experiments are presented in Section 2.6. Mathematical proofs and additional numerical details are available in Section 2.7.

2.2 Preliminaries on Monte Carlo integration

The aim of this section is to present the required mathematical framework for Monte Carlo integration and the variance reduction methods of interest, namely adaptive importance sampling and the control variate technique. Recall that $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is an integrand and p a probability density on $\mathbb{R}^d$. The aim is to compute $E_p[f] = \int f p \, d\lambda$.

Adaptive importance sampling. In adaptive importance sampling (AIS), $E_p[f]$ is estimated by a weighted mean over a sample of random particles $X_1, \dots, X_n$ in $\mathbb{R}^d$. Since appropriate sampling densities naturally depend on f and p , we generally cannot simulate from them. They are then approximated in an adaptive manner by a family of tractable densities $(q_i)_{i \geq 0}$ that often evolve towards a density q_{opt} that optimizes some criterion. While the starting density q_0 is fixed, the density q_i for $i \geq 1$ is determined in function of the particles $X_1, \dots, X_i$ already sampled; think for instance of a parametric family, where the parameter of q_i is a function of $X_1, \dots, X_i$. Given the particles $X_1, \dots, X_i$, the next particle, X_{i+1} , is then drawn from q_i . Formally, let $(X_i)_{i \geq 1}$ be a sequence of random vectors on $\mathbb{R}^d$ defined on some probability space $(\Omega, \mathcal{F}, P)$. The distribution of the sequence $(X_i)_{i \geq 1}$ is specified by its policy as defined below.

Definition 2.1 (Policy). A policy is a random sequence of probability density functions $(q_i)_{i \geq 0}$ on $\mathbb{R}^d$ adapted to the σ -field $(\mathcal{F}_i)_{i \geq 0}$ defined by $\mathcal{F}_0 = \{\emptyset, \Omega\}$ and $\mathcal{F}_i = \sigma(X_1, \dots, X_i)$ for $i \geq 1$. The sequence $(q_i)_{i \geq 0}$ is the policy of $(X_i)_{i \geq 1}$ whenever X_i has density q_{i-1} conditionally on $\mathcal{F}_{i-1}$.

The (normalized) adaptive importance sampling estimate of $E_p[f]$ is then defined as

$$I_n^{(\text{ais})}(f) = \frac{\sum_{i=1}^n w_i f(X_i)}{\sum_{i=1}^n w_i} \quad \text{where} \quad w_i = \frac{p(X_i)}{q_{i-1}(X_i)} \quad \text{for } i = 1, \dots, n. \quad (2.1)$$

The sampling weights w_i reflect the fact that X_i has been sampled from q_{i-1} rather than from p . The division by $\sum_{i=1}^n w_i$ rather than by n has two benefits: first, the integration is exact for constant integrands, and second, p needs to be known only up to a proportionality constant, an advantage for Bayesian inference.

Since updating the density q_i at each iteration may be computationally expensive, it is customary to hold it fixed over a pre-determined number of iterations. Writing $n = n_1 + \dots + n_T$ in terms of positive integers $(n_t)_{t=1}^T$ called the *allocation policy*, the AIS estimate then becomes

$$I_T^{(\text{ais})}(f) = \frac{\sum_{t=1}^T \sum_{i=1}^{n_t} w_{t,i} f(X_{t,i})}{\sum_{t=1}^T \sum_{i=1}^{n_t} w_{t,i}} \quad \text{where} \quad w_{t,i} = \frac{p(X_{t,i})}{q_t(X_{t,i})} \quad (2.2)$$

for $t = 1, \dots, T$ and $i = 1, \dots, n_t$. At stage t , the particles $X_{t,1}, \dots, X_{t,n_t}$ are sampled independently from q_{t-1} , while all particles sampled up to and including stage t are used to determine the sampling density q_t for stage $t+1$. It is easy to see that the two formulations of the AIS estimate are equivalent: (2.1) arises from (2.2) by setting $n_t = 1$ for all t , while (2.2) can be obtained from (2.1) by constructing the policy in such a way that the densities q_i do not change within integer intervals of the form $\{0, \dots, n_1-1\}$, $\{n_1, \dots, n_1+n_2-1\}$, and so on. While the shorter representation (2.1) is more convenient for theoretical purposes, formulation (2.2) is the one used in practice (see Section 2.6).

Interestingly, the AIS estimate (2.1) may be seen as a weighted least-squares estimate minimizing the loss function $a \mapsto \sum_{i=1}^n w_i (f(X_i) - a)^2$. This perspective is key to understand control variates.

Control variates. The control variates method is a variance reduction technique that consists in incorporating a new piece of information—the known values of the integrals of some control functions—in a basic Monte Carlo framework. Control variates are simply functions $\varphi_1, \dots, \varphi_m \in L_2(p)$ with known integrals. Without loss of generality, assume that $E_p[\varphi_j] = 0$ for all $j = 1, \dots, m$. Let $\varphi = (\varphi_1, \dots, \varphi_m)^\top$ denote the $\mathbb{R}^m$ -valued function with the m control variates as elements. For any coefficient vector $\beta \in \mathbb{R}^m$, we have $E_p[f - \beta^\top \varphi] = E_p[f]$. Given an independent random sample $X_1, \dots, X_n$ from p , any $\beta \in \mathbb{R}^m$ therefore results in an unbiased estimator of $E_p[f]$ by

$$I_n^{(\text{cv})}(f, \beta) = \frac{1}{n} \sum_{i=1}^n (f(X_i) - \beta^\top \varphi(X_i)). \quad (2.3)$$

Provided the $m \times m$ covariance matrix $G = E_p[\varphi \varphi^\top]$ is invertible, there is a unique coefficient vector $\beta^* \in \mathbb{R}^m$ for which the variance of $I_n^{(\text{cv})}(f)$ is minimal and it is given by

$$\beta^* = (E_p[\varphi \varphi^\top])^{-1} E_p[\varphi f]. \quad (2.4)$$

This vector being generally unknown, it needs to be estimated from the particles $X_1, \dots, X_n$. Casting the problem in an ordinary least squares framework leads to the control variate estimate

$$I_n^{(\text{cv})}(f) = I_n^{(\text{cv})}(f, \hat{\beta}_n^{(\text{cv})}) = \hat{\alpha}_n^{(\text{cv})} \quad \text{where} \quad (\hat{\alpha}_n^{(\text{cv})}, \hat{\beta}_n^{(\text{cv})}) \in \arg \min_{(a,b) \in \mathbb{R} \times \mathbb{R}^m} \frac{1}{n} \sum_{i=1}^n (f(X_i) - a - b^\top \varphi(X_i))^2. \quad (2.5)$$

The estimator $I_n^{(\text{cv})}(f)$ is well-defined provided the minimizer $\hat{\alpha}_n^{(\text{cv})}$ to (2.5) is unique. This is the case if and only if there does not exist $b \in \mathbb{R}^m$ such that $b^\top \varphi(X_i) = 1$ for all $i = 1, \dots, n$.

The asymptotic distribution of $I_n^{(\text{cv})}(f)$ as $n \rightarrow \infty$ is the same as if the variance-minimizing vector β^* were used in (2.3). In particular, the asymptotic variance of $I_n^{(\text{cv})}(f)$ is $\sigma_m^2(f)/n$ where

$$\sigma_m^2(f) = \min_{\beta \in \mathbb{R}^m} \mathbb{E}_p[(f - \mathbb{E}_p[f] - \beta^\top \varphi)^2].$$

Interestingly, when using only the first ℓ out of m control variates, where $\ell \in \{0, 1, \dots, m\}$, we have $\sigma_m^2(f) \leq \sigma_\ell^2(f)$. In terms of asymptotic variance, it therefore never harms to add more control variates. Their construction will be addressed in Section 2.5.1.

2.3 Combining adaptive importance sampling with control variates

AISCV estimator. Consider the same integration problem $\mathbb{E}_p[f] = \int f p \, d\lambda$ as in Section 2.2. With the idea of performing variance reduction when calculating integrals with respect to the posterior density in Bayesian inference, we incorporate control variates into the AIS estimate. Let the particles $(X_i)_{i \geq 1}$ be generated according to a policy $(q_i)_{i \geq 0}$ as in Definition 2.1. Let $\varphi = (\varphi_1, \dots, \varphi_m)^\top$ be a vector of control variates, i.e., $\varphi_j \in L_2(p)$ and $\mathbb{E}_p[\varphi_j] = 0$ for every $j = 1, \dots, m$. Combining (2.1) and (2.3), the proposed estimate takes the form

$$I_n^{(\text{aiscv})}(f, \beta) = \frac{\sum_{i=1}^n w_i (f(X_i) - \beta^\top \varphi(X_i))}{\sum_{i=1}^n w_i}, \quad (2.6)$$

where $\beta \in \mathbb{R}^m$ remains to be determined. To do so, the ordinary least-squares problem in (2.5) is replaced by a weighted one, yielding the novel AISCV estimator

$$I_n^{(\text{aiscv})}(f) = I_n^{(\text{aiscv})}(f, \hat{\beta}_n) = \hat{\alpha}_n \quad \text{where} \quad (\hat{\alpha}_n, \hat{\beta}_n) \in \arg \min_{(a,b) \in \mathbb{R} \times \mathbb{R}^m} \sum_{i=1}^n w_i (f(X_i) - a - b^\top \varphi(X_i))^2. \quad (2.7)$$

The estimator is well-defined only if the minimizer $\hat{\alpha}_n$ is unique—the minimizer $\hat{\beta}_n$ need not be. We will come back to this in the next paragraph.

As in (2.2), the policy may be divided into T stages in order to reduce the number of times the sampler needs to be updated. Stage $t = 1, \dots, T$ has length n_t , with $\sum_{t=1}^T n_t = n$. Within each stage, the sampling density remains constant. In practice, this leads to the AISCV estimate in Algorithm 2.1.

Algorithm 2.1 Adaptive Importance Sampling with Control Variates (AISCV)

Require: integrand f , target density p (up to a proportionality constant), number of stages

$T \in \mathbb{N}^*$, allocation policy $(n_t)_{t=1}^T$, initial density q_0 , update rule for the sampling policy

- 1: **for** $t = 1, \dots, T$ **do**
 - 2: Generate an independent random sample $X_{t,1}, \dots, X_{t,n_t}$ from q_{t-1}
 - 3: Compute the vector of weights $(w_{t,i})_{i=1}^{n_t}$ where $w_{t,i} = p(X_{t,i})/q_{t-1}(X_{t,i})$
 - 4: Construct the matrix of control variates $\Phi_t = (\varphi_j(X_{t,i}))_{i=1, \dots, n_t}^{j=1, \dots, m}$
 - 5: Evaluate the integrand in the particles: $(f(X_{t,i}))_{i=1}^{n_t}$
 - 6: Update the sampler q_t based on all previous particles $(X_{s,i} : s = 1, \dots, t; i = 1, \dots, n_s)$
 - 7: **end for**
 - 8: Compute $(\hat{\alpha}_T, \hat{\beta}_T) = \arg \min_{(a,b) \in \mathbb{R} \times \mathbb{R}^m} \left\{ \sum_{t=1}^T \sum_{i=1}^{n_t} w_{t,i} (f(X_{t,i}) - a - b^\top \varphi(X_{t,i}))^2 \right\}$
 - 9: **return** $I_n^{(\text{aiscv})}(f) = \hat{\alpha}_T$.
-

Quadrature rule. The AIS estimate (2.1) is a quadrature rule with quadrature points X_i and quadrature weights proportional to the sampling weights w_i . The AISCV estimate (2.7) has the same property, but with adapted quadrature weights. Let $e_n = (e_{n,i})_{i=1, \dots, n}$ be the vector of residuals resulting from the weighted least-squares regression of the constant vector $\mathbf{1}_n = (1, \dots, 1)^\top \in \mathbb{R}^n$ on the control variates but without intercept:

$$e_{n,i} = 1 - \hat{\beta}_n(\mathbf{1}_n)^\top \varphi(X_i) \quad \text{where} \quad \hat{\beta}_n(\mathbf{1}_n) \in \arg \min_{b \in \mathbb{R}^m} \sum_{i=1}^n w_i (1 - b^\top \varphi(X_i))^2. \quad (2.8)$$

Even though the vector $\hat{\beta}_n(\mathbf{1}_n)$ is not necessarily unique, the weighted least squares fit $(\hat{\beta}_n(\mathbf{1}_n)^\top \varphi(X_i))_{i=1, \dots, n}$ always is. According to the next proposition, the quadrature weights are proportional to $(w_i e_{n,i})_{i=1, \dots, n}$.

Proposition 2.1 (AISCV quadrature rule). *The minimizer $\hat{\alpha}_n$ in (2.7) is unique if and only if $e_n \neq 0$ in (2.8). In that case, the AISCV estimate is*

$$I_n^{(\text{aiscv})}(f) = \hat{\alpha}_n = \frac{\sum_{i=1}^n w_i e_{n,i} f(X_i)}{\sum_{i=1}^n w_i e_{n,i}}. \quad (2.9)$$

If $e_n = 0$, then there exists $b \in \mathbb{R}^m$ such that $b^\top \varphi(X_i) = 1$ for all $i = 1, \dots, n$. In that case, the minimizer $\hat{\alpha}_n$ in (2.7) is not unique and the AISCV estimate is not well-defined. To remedy this, one can for instance reduce the number of control variates. This issue already occurs with the ordinary control variate estimator in (2.3).

Rather than requiring a different weighted least squares problem for every integrand f as in (2.7), the quadrature rule in (2.9) only involves a single weighted least squares problem (2.8), whatever f . Given the quadrature weights, calculating the AISCV estimate for a novel integrand only requires the evaluations of that function on the sampled particles, making the whole procedure a *post-hoc* scheme. The steps in case the sampling policy is divided into T stages are given in Algorithm 2.2, which gives the same result as Algorithm 2.1, but with less effort if multiple integrands f are into play.

Algorithm 2.2 Quadrature Rule – AISCV *post-hoc* scheme

Require: integrand f , $T \in \mathbb{N}^*$, allocation policy $(n_t)_{t=1}^T$, weights $(w_t)_{t=1}^T$ with $w_t = (w_{t,i})_{i=1}^{n_t}$, matrices $(\Phi_t)_{t=1}^T$ with $\Phi_t = (\varphi_j(X_{t,i}))_{i=1, \dots, n_t}^{j=1, \dots, m}$, particles $(X_{t,i} : t = 1, \dots, T; i = 1, \dots, n_t)$

- 1: Compute $\hat{\beta}_n(\mathbf{1}_n) = \arg \min_{b \in \mathbb{R}^m} \sum_{t=1}^T \sum_{i=1}^{n_t} w_{t,i} (1 - b^\top \varphi(X_{t,i}))^2$
- 2: Compute $u_t = \text{diag}(w_t) [\mathbf{1}_{n_t} - \Phi_t \hat{\beta}_n(\mathbf{1}_n)]$ for $t = 1, \dots, T$
- 3: Compute $s = \sum_{t=1}^T \sum_{i=1}^{n_t} u_{t,i}$
- 4: Compute weights $v_{t,i} = u_{t,i}/s$ for $t = 1, \dots, T$ and $i = 1, \dots, n_t$
- 5: **return** $I_T^{(\text{aiscv})}(f) = \sum_{t=1}^T \sum_{i=1}^{n_t} v_{t,i} f(X_{t,i})$

2.4 Theoretical properties of the AISCV estimate

Here we point out several theoretical properties of the novel AISCV estimate. A first point is that the integration rule is exact on the linear span of the control variates and the constant function.

Proposition 2.2 (Exact integration). *For integrands of the form $f = \alpha + \beta^\top \varphi$ for $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}^m$, the AISCV estimate is exact: $I_n^{(\text{aiscv})}(f) = \alpha = \mathbb{E}_p[f]$.*

A second property is that we may apply arbitrary invertible linear transformations to the control variates without changing the AISCV estimate. This can be advantageous computationally, to make the underlying weighted least squares problem more stable numerically. Also, it means that without loss of generality, we may assume that the control variates are uncorrelated and have unit variance, which simplifies the theoretical performance analysis.

Proposition 2.3 (Invariance). *If the matrix $A \in \mathbb{R}^{m \times m}$ is invertible, then the AISCV estimate based on the control variates $A\varphi$ is the same as the one based on φ .*

Our main result is a non-asymptotic bound on the error of the AISCV estimate for $\int f p \, d\lambda$ when $\int f^2 p \, d\lambda$ is finite. First, we introduce some assumptions and definitions.

The first condition that is required concerns the policy given by the AIS part of the algorithm. It is supposed that any element from the policy should dominate the function p .

Assumption 1 (Dominated measures). *There exists $c \geq 1$ such that, for all $x \in \mathbb{R}^d$ and for any $i = 1, \dots, n$, we have $p(x) \leq c \cdot q_i(x)$.*

This assumption represents a *safe* approach to importance sampling, as the policy will always allow to sample in places where p is positive. A well-known and well-spread (Hesterberg, 1995; Owen and Zhou, 2000; Delyon and Portier, 2021) technique to achieve such a defensive strategy is to use mixture density $q_i = (1 - \eta)p_i + \eta q_0$ where $\eta \in (0, 1)$ and where q_0 has sufficiently heavy tails to dominate p . Such a mixture allows to choose the densities p_i with some flexibility using in principle any AIS algorithm. Second, the control variates shall be linearly independent and bounded.

Assumption 2 (Control variates). *We have $\sup_{x: p(x) > 0} |\varphi_j(x)| < \infty$ for all $j = 1, \dots, m$. The matrix $G = \int \varphi \varphi^\top p \, d\lambda$ is invertible.*

The previous condition allows to define the standardized vector of control variates as $\bar{\varphi} = G^{-1/2} \varphi$. By Proposition 2.3, this change does not affect the AISCV estimate. The orthonormal control variates $\bar{\varphi}$ will play a key role through the following quantity

$$B = \sup_{x: p(x) > 0} \|\bar{\varphi}(x)\|_2^2.$$

The quadratic form $\|\bar{\varphi}(x)\|_2^2 = \varphi(x)^\top G^{-1} \varphi(x)$ is referred to as the *leverage function* in ordinary linear regression as it quantifies the influence of a training point x on the prediction of the observed response. It is invariant with respect to invertible linear transformations of the control variate vector.

Assumption 2 and the fact that the integrand f is square integrable with respect to p allows to define the residual function $\varepsilon = f - \int f p \, d\lambda - \varphi^\top \beta^*$ where β^* has been introduced in (2.4) as a minimizer of the residual variance. Since we work in the space $L_2(p)$, we assume without loss of generality that f and φ vanish outside $\{x : p(x) > 0\}$ and we put $\varepsilon(x) = 0$ for $x \in \mathbb{R}^d$ such that $p(x) = 0$. The residual function ε should satisfy the following tail condition.

Assumption 3 (Residual tail). *There exists $\tau > 0$ such that, for all $t > 0$ and all integer $i \geq 1$, we have $P[|w_i \varepsilon(X_i)| > t \mid \mathcal{F}_{i-1}] \leq 2 \exp(-t^2/(2\tau^2))$.*

The previous assumption concerns both the function ε and the policy sequence $(q_i)_{i \geq 0}$. Since $\mathbb{E}[w_i \varepsilon(X_i) \mid \mathcal{F}_{i-1}] = 0$, it is implied by the so-called sub-Gaussian condition (Boucheron et al., 2013) that $\mathbb{E}[\exp(\lambda w_i \varepsilon(X_i)) \mid \mathcal{F}_{i-1}] \leq \exp(-\lambda^2 \tau^2/2)$ for any $\lambda \in \mathbb{R}$. In the proof of Theorem 2.1, Assumption 3 allows to derive concentration bounds on residual-based sums using recent results from Jin et al. (2019); Leluc et al. (2021). We are now in position to state our main result on the error of the AISCV estimate.

Theorem 2.1 (Concentration inequality for AISCV estimate). *If Assumptions 1, 2 and 3 hold, then, for any $\delta \in (0, 1)$ and for all $n \geq C_1 c^2 B \log(10m/\delta)$, we have, with probability at least $1 - \delta$, that*

$$\left| I_n^{(\text{aiscv})}(f) - \int_{\mathbb{R}^d} f(x) p(x) dx \right| \leq C_2 \tau \sqrt{\frac{\log(10/\delta)}{n}} + C_3 c B \tau \frac{\log(10m/\delta)}{n},$$

where C_1, C_2, C_3 are universal constants specified in the proof.

Remark 2.1 (Understanding τ). *The quantity τ in Assumption 3 is related to the conditional variance $\mathbb{E}[w_i^2 \varepsilon^2(X_i) \mid \mathcal{F}_{i-1}]$. They actually coincide when $w_i \varepsilon(X_i)$ is Gaussian. For a policy satisfying Assumption 1, $\mathbb{E}[w_i^2 \varepsilon^2(X_i) \mid \mathcal{F}_{i-1}] \leq c \sigma_m^2$ which for certain combinations of integrands and control functions scales as $m^{-s/d}$ (Portier and Segers, 2019) where the parameter s represents the degree of smoothness of f .*

Remark 2.2 (Convergence rates). *Consider an asymptotic regime where the number of control variates m tends to infinity with the sample size n . The AISCV estimate improves upon the AIS method ($m = 0$), which has rate $1/\sqrt{n}$, as soon as $\tau + \tau B \log(m)/\sqrt{n} \rightarrow 0$. To recover the same order of an oracle estimate with rate $\tau/\sqrt{n}$, one must have $B \log(m) = O(\sqrt{n})$ as $n \rightarrow \infty$.*

2.5 Practical considerations

This section presents ways to build control variates using either families of polynomials or general functions based on Stein's method, with a highlight on computations in the Bayesian framework.

2.5.1 Control variate constructions

Orthogonal polynomials. When the target density p can be decomposed as a product of univariate densities $p = \rho_1 \otimes \cdots \otimes \rho_d$, multidimensional control functions may be constructed based on univariate ones. This happens for

instance for the uniform distribution over the unit cube $[0, 1]^d$ or with uncorrelated Gaussian distributions on $\mathbb{R}^d$. Such univariate control variates may be easily constructed using families of polynomials (Gautschi, 2004), such as Legendre polynomials for the uniform distribution on $[0, 1]$ and Hermite polynomials for the Gaussian distribution on $\mathbb{R}$. This technique can also be used when p is dominated by another density p^* having the said product form by transforming zero-mean control variates φ^* with respect to p^* via $\varphi = \varphi^* p^* / p$.

Let $(\varphi_1, \dots, \varphi_k)$ be a vector of univariate control functions with respect to a density ρ , i.e., $E_\rho[\varphi_j] = 0$ for all $j = 1, \dots, k$. Let $\varphi_0 = 1$ denote the constant function equal to one. For a multi-index $\ell = (\ell_1, \dots, \ell_d)$ in $\{0, \dots, k\}^d \setminus \{(0, \dots, 0)\}$, multivariate controls with respect to $\rho^{\otimes d}$ are built by forming tensor products of the form $\varphi_\ell(x_1, \dots, x_d) = \varphi_{\ell_1}(x_1) \cdots \varphi_{\ell_d}(x_d)$, yielding a total number of $m = (k+1)^d - 1$ control functions. Alternative approaches yielding smaller control spaces consist of imposing $\ell_j = 0$ for all but a small number of coordinates $j = 1, \dots, d$ or by the constraint $\ell_1 + \dots + \ell_d \leq Q$ for some $Q \geq 1$.

Stein control variates. In the general case where one has only access to the evaluations of p , control variates may be constructed using Stein's method. The technique relies on the gradient $\nabla_x \log p(x)$ which can either be directly computed (see the example of Bayesian regression below) or which may be available through automatic differentiation provided in popular APIs such as Tensorflow and PyTorch (Abadi et al., 2016; Paszke et al., 2017). Let $\Delta_x = \nabla_x^\top \nabla_x$ denote the Laplace operator. By definition, the second-order Stein operator $\mathcal{L}$ (Stein, 1972; Gorham and Mackey, 2015) associated to the density p is defined by:

$$\forall \psi \in C^2(\mathbb{R}^d, \mathbb{R}), \quad (\mathcal{L}\psi)(x) = \Delta_x \psi(x) + \nabla_x \psi(x)^\top \nabla_x \log p(x).$$

The transformation guarantees that $E_p[\mathcal{L}\psi] = 0$ for all ψ with weak regularity conditions (Mira et al., 2013). Therefore, we can build infinitely many control variates $\varphi_\psi = \mathcal{L}\psi$ from given functions ψ . One simple way is to let ψ be a polynomial with bounded total degree: for a degree vector $\alpha = (\alpha_1, \dots, \alpha_d) \in \mathbb{N}^d$ with $\alpha_1 + \dots + \alpha_d \leq Q$, define $\psi_\alpha(x) = x_1^{\alpha_1} \cdots x_d^{\alpha_d}$. Given the dimension d and the total degree Q , there are $m = \binom{d+Q}{d} - 1$ such degree vectors, yielding the associated control variates $\varphi_\alpha = \varphi_{\psi_\alpha}$. For fast computation, note that, writing $\phi_\alpha(x) = \psi_\alpha(x) \mathbb{1}_d$, $D_1(x) = \text{diag}(\alpha_1/x_1, \dots, \alpha_d/x_d)$ and $D_2(x) = \text{diag}(\alpha_1(\alpha_1 - 1)/x_1^2, \dots, \alpha_d(\alpha_d - 1)/x_d^2)$, we have $\nabla_x \psi_\alpha(x) = D_1(x) \phi_\alpha(x)$ and $\Delta_x \psi_\alpha(x) = \mathbb{1}_d^\top (D_2(x) \phi_\alpha(x))$. In practice, all combinations of α are stored in a matrix $A \in \mathbb{N}^{m \times d}$.

2.5.2 Bayesian inference

Given data $\mathcal{D}$ and a parameter of interest $\theta \in \Theta \subset \mathbb{R}^d$, posterior integrals take the form $\int_{\mathbb{R}^d} f(\theta) p(\theta|\mathcal{D}) d\theta$, where $p(\theta|\mathcal{D}) \propto \ell(\mathcal{D}|\theta)\pi(\theta)$ is the posterior distribution, proportional to a prior $\pi(\theta)$ and a likelihood function $\ell(\mathcal{D}|\theta)$. For instance, when $f(\theta) = \theta$, the integral above recovers the posterior mean. Stein control variates involve the computation of the gradient of the log-posterior $\nabla_{\theta} \log p(\theta|\mathcal{D})$, which implicitly relies on the score function $\nabla_{\theta} \log \ell(\mathcal{D}|\theta)$. We point out two common examples—linear and logistic regression—where these functions are easy to compute.

Bayesian linear regression. Consider a linear regression problem comprised of observations $X \in \mathbb{R}^{N \times d}$ with labels $y \in \mathbb{R}^N$. In the Gaussian fixed design setting, the predictor x_i produces the response $y_i = x_i^\top \theta + \varepsilon_i$ where $\varepsilon_1, \dots, \varepsilon_N \sim \mathcal{N}(0, \sigma^2)$ are centered Gaussian noises. The likelihood $\ell(X, y|\theta)$ is proportional to $(\sigma^2)^{-N/2} \exp(-(y - X\theta)^\top (y - X\theta)/(2\sigma^2))$, yielding the score function $\nabla_{\theta} \log \ell(X, y|\theta) = X^\top (y - X\theta)/(\sigma^2)$.

Bayesian logistic regression. Next, consider the logistic regression problem comprised of observations $X \in \mathbb{R}^{N \times d}$ with associated binary labels $y \in \{0, 1\}^N$. Letting $\sigma(s) = 1/(1 + e^{-s})$ denote the sigmoid function, the likelihood function is $\ell(X, y|\theta) = \prod_{i=1}^N \sigma(\theta^\top x_i)^{y_i} (1 - \sigma(\theta^\top x_i))^{1-y_i}$. The score function is simply $\nabla_{\theta} \log \ell(X, y|\theta) = X^\top (y - \sigma(X\theta))$.

2.6 Numerical illustration

To compare the finite-sample performance of the AIS and AISCV estimators, we first present in Section 2.6.1 synthetic data examples involving the integration problem over the unit cube $[0, 1]^d$ and then with respect to some Gaussian mixtures as in Cappé et al. (2008). The goal is to compute $\int f p d\lambda$ for vectors of integrands $f : \mathbb{R}^d \rightarrow \mathbb{R}^p$. We consider various dimensions $d > 1$ and several choices for the number of control variates m . Section 2.6.2 deals with real-world datasets in the context of Bayesian inference.

Parameters. In all simulations, the sampling policy is taken within the family of multivariate Student t distributions of degree ν denoted by $\{q_{\mu, \Sigma_0} : \mu \in \mathbb{R}^d\}$ with $\Sigma_0 = \sigma_0 I_d (\nu - 2)/\nu$ and $\nu > 2, \sigma_0 > 0$. Similarly to Portier and Delyon (2018), the mean μ_t is updated at each stage $t = 1, \dots, T$ by the generalized method of moments (GMM), leading to $\mu_t = \frac{\sum_{s=1}^t \sum_{i=1}^{n_s} w_{s,i} X_{s,i}}{\sum_{s=1}^t \sum_{i=1}^{n_s} w_{s,i}}$. The allocation policy is fixed to $n_t = 1000$ and the number of stages is $T \in \{5; 10; 20; 30; 50\}$. The different Monte Carlo estimates are compared by their mean squared error (MSE) obtained over 100 independent replications.

2.6.1 Synthetic examples

Integration on $[0, 1]^d$. We seek to integrate functions f with respect to the uniform density $p(x) = 1$ for $x \in [0, 1]^d$ in dimensions $d \in \{4; 8\}$. We rely on Legendre polynomials for the control variates. Consider the integrands $f_1(x) = 1 + \sin(\pi(2d^{-1} \sum_{i=1}^d x_i - 1))$, $f_2(x) = \prod_{i=1}^d (2/\pi)^{1/2} x_i^{-1} e^{-\log(x_i)^2/2}$ and $f_3(x) = \prod_{i=1}^d \log(2) 2^{1-x_i}$, all of which integrate to 1 on $[0, 1]^d$. None of the integrands is a linear combination of the control variates. The policy parameters are $\mu_0 = (0.5, \dots, 0.5)^\top \in \mathbb{R}^d$, $\nu = 8$, and $\sigma_0 = 0.1$. The control variates are built out of tensor products of Legendre polynomials where the degree ℓ_j equals 0 for all but two coordinates, leading to a total number of $m = kd + k^2d(d-1)/2$ control variates. The maximum degree in each variable is $k = 6$, yielding $m = 240$ and $m = 1056$ control variates in dimensions $d = 4$ and $d = 8$ respectively. Figure 2.1 presents the boxplots of the AIS and AISCV estimates. The error reduction obtained thanks to the control variates is huge: the AISCV estimate has a mean squared error smaller than the one of the AIS estimate by a factor at least 10 and up to 100 (see Table 2.1 in the Appendix).

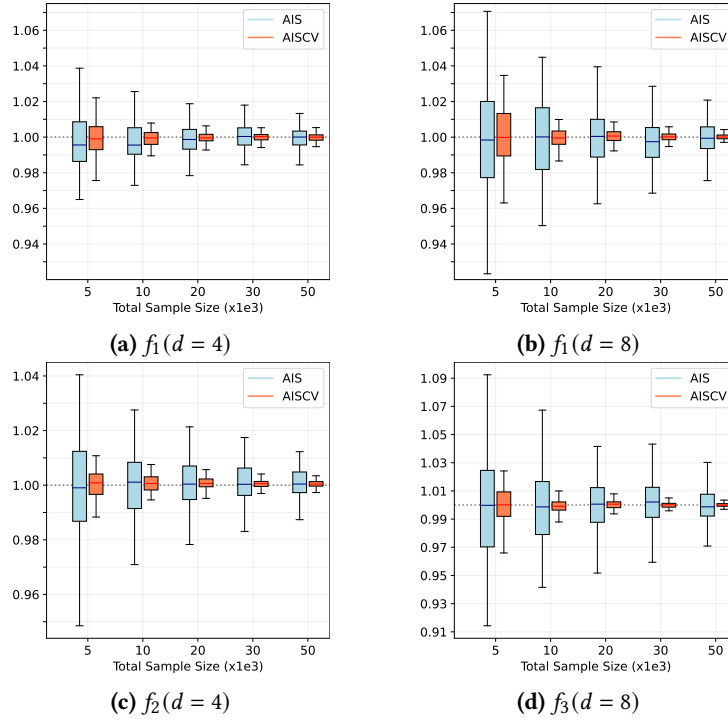


Figure 2.1: Integration on $[0, 1]^d$: boxplots of estimates $I_n^{(\text{ais})}(f)$ and $I_n^{(\text{aiscv})}(f)$ with integrands f_1, f_2, f_3 in dimensions $d \in \{4; 8\}$ obtained over 100 replications. The true integral equals 1.

Gaussian mixture p and Stein control variates. In this setting we assume we only have access to the evaluations of the target density p . We consider the classical example introduced in Cappé et al. (2008) where p is a mixture of two Gaussian distributions. The control variates are built using Stein's method (Section 2.5.1) out of polynomials of total degree at most $Q \in \{2; 3\}$, leading to a number of control variates $m \in \{14; 34\}$ in dimension $d = 4$ and $m \in \{44; 164\}$ in dimension $d = 8$ respectively. We consider two cases: an isotropic and an anisotropic one.

Isotropic case. Let $p_\Sigma(x) = 0.5\Phi_\Sigma(x - \mu) + 0.5\Phi_\Sigma(x + \mu)$ where $\Sigma = I_d/d$, $\mu = (1, \dots, 1)^\top / (2\sqrt{d})$ and Φ_Σ is the multivariate normal density function with zero mean and covariance matrix Σ . The Euclidean distance between the two mixture centers is 1, independently of d . The initial density q_0 is the multivariate Student t distribution with mean $(1, -1, 0, \dots, 0)/\sqrt{d}$ and variance $(5/d)I_d$. The initial mean value differs from the null vector to prevent the naive algorithm using the initial density from having good results due to the symmetrical set-up.

Anisotropic case. In this case, the mixture is unbalanced and each Gaussian is anisotropic. The target density is $p_V(x) = 0.75\Phi_V(x - \mu) + 0.25\Phi_V(x + \mu)$ where $\mu = (1, \dots, 1)^\top / (2\sqrt{d})$ and $V = \text{diag}(10, 1, \dots, 1)/d$. The initial density q_0 is the same as for the isotropic case.

Figure 2.2 presents the evolution of the logarithm of the mean squared error $\|\hat{I}(f) - I(f)\|_2^2$. Once again, the AISCV estimators are the clear winners with a mean squared error smaller by a factor up to 1000 for the anisotropic case (see Table 2.2 in the Appendix).

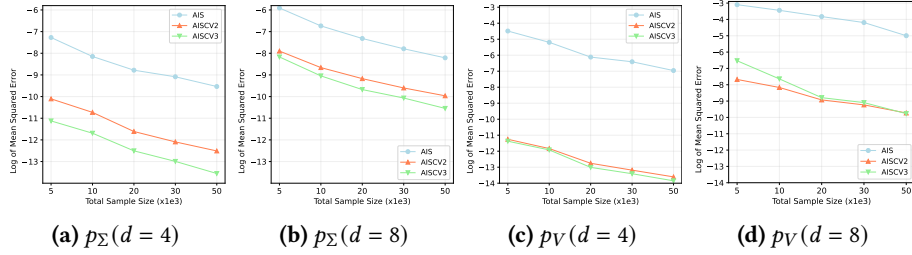


Figure 2.2: Gaussian mixture density: Logarithm of $\|\hat{I}(f) - I(f)\|_2^2$ for $f(x) = x$ with target isotropic p_Σ and anisotropic p_V in dimensions $d \in \{4; 8\}$ obtained over 100 replications.

2.6.2 Real-world examples

We place ourselves in the framework of Bayesian linear regression (Section 2.5.2) with features $X \in \mathbb{R}^{N \times d}$ and continuous responses $y \in \mathbb{R}^N$. The pos-

terior distribution $p(\theta|\mathcal{D})$ involves a Gaussian prior $\pi(\theta) \sim \mathcal{N}(\mu_a, \Sigma_a)$ and a likelihood function $\ell(\mathcal{D}|\theta)$ proportional to $(\sigma^2)^{-N/2} \exp(-(y - X\theta)^\top (y - X\theta)/(2\sigma^2))$. The noise level is fixed and taken sufficiently large at $\sigma = 50$ to account for general priors. The posterior distribution is Gaussian too: $\mathcal{N}(\mu_b, \Sigma_b)$ with $\mu_b = \Sigma_b(\sigma^{-2}X^\top y + \Sigma_a^{-1}\mu_a)$ and $\Sigma_b = (\sigma^{-2}X^\top X + \Sigma_a^{-1})^{-1}$. The integrand is $f(\theta) = \sum_{i=1}^d \theta_i^2$ and the control variates are built with the Stein operator (Section 2.5.1) out of monomials with total degree $Q \in \{1, 2\}$, leading to the AISCV1 and AISCV2 estimators respectively.

Datasets and parameters. Classical datasets from Dua and Graff (2019) are considered : *housing* ($N = 506; d = 13; m \in \{12; 104\}$); *abalone* ($N = 4177; d = 8; m \in \{7; 44\}$); *red wine* ($N = 1599; d = 11; m \in \{10; 77\}$); and *white wine* ($N = 4898; d = 11; m \in \{10; 77\}$). The initial density is the multivariate Student t distribution with $\nu = 10$ degrees of freedom, zero mean and covariance matrix Σ_b .

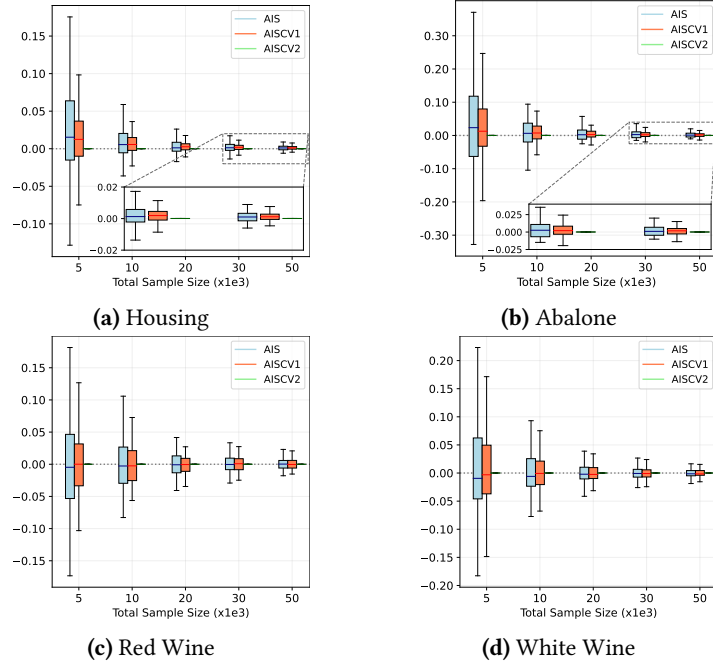


Figure 2.3: Bayesian linear regression: boxplots of $(\hat{I}(f) - I(f))/I(f)$ for $f(\theta) = \sum_{j=1}^d \theta_j^2$.

Results. Figure 2.3 presents the boxplots of the relative error $\frac{\hat{I}(f) - I(f)}{I(f)}$, revealing the benefits of control variates even with polynomials of degree $Q = 1$. When $Q = 2$, the error of the AISCV2 estimate is virtually zero (see Table 2.3 in the Appendix), in line with Proposition 2.2. The mean squared error of the AISCV1 estimate is smaller than that of the AIS estimate by a factor ranging

between 2 and 10.

2.7 Appendix

Technical Lemmas and auxiliary results are provided in Section 2.7.1. Section 2.7.2 collects additional theoretical properties of the AISCV estimator while the technical proofs of the propositions and main theorem are presented in Section 2.7.3. Finally, Section 2.7.4 presents additional numerical values associated to the numerical experiments on synthetic examples and real-world datasets for Bayesian linear regression.

2.7.1 Auxiliary results

Lemmas on (Random) Matrices inequalities

Definition 2.2. Let A and Ψ be Hermitian matrices of the same dimension. We say that $A \leq \Psi$ if and only if $\Psi - A$ is positive semidefinite.

Definition 2.3 (Tropp (2015), Definition 2.1.2). Let $f : I \rightarrow \mathbb{R}$ where I is an interval of the real line. Consider a $d \times d$ Hermitian matrix A whose eigenvalues are contained in I . Define a $d \times d$ Hermitian matrix $f(A)$, called the standard matrix function, using an eigenvalue decomposition of A , by

$$f(A) = Q \begin{bmatrix} f(\lambda_1) & & \\ & \ddots & \\ & & f(\lambda_d) \end{bmatrix} Q^* \quad \text{where} \quad A = Q \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_d \end{bmatrix} Q^*.$$

Remark 2.3. The matrix exponential e^A and the matrix logarithm $\log(A)$ are the standard matrix functions.

Lemma 2.1 (Tropp (2015), Example 8.3.4). The trace exponential map is monotone:

$$A \leq \Psi \text{ implies } \operatorname{tr} e^A \leq \operatorname{tr} e^\Psi$$

for all Hermitian matrices A and Ψ .

Lemma 2.2 (Tropp (2015), Proposition 3.2.1). For any random Hermitian matrix Y , for all $t \in \mathbb{R}$, we have

$$\mathbb{P}(\lambda_{\min}(Y) \leq t) \leq \inf_{\theta < 0} e^{-\theta t} \mathbb{E}[\operatorname{tr}(e^{\theta Y})].$$

Lemma 2.3 (Tropp (2015), Lemma 5.4.1). Assume that A is a random matrix with $0 \leq \lambda_{\min}(A)$ and, for some constant $L > 0$, $\lambda_{\max}(A) \leq L$. Then, for all $\theta \in \mathbb{R}$,

$$\log(\mathbb{E}[e^{\theta A}]) \leq \eta(\theta) \mathbb{E}[A], \quad \eta(\theta) = L^{-1}(e^{\theta L} - 1).$$

Lemma 2.4 (Tropp (2015), Corollary 3.4.2). *Let Ψ be a fixed Hermitian matrix and A a random Hermitian matrix of the same dimension. Then*

$$\mathbb{E} [\text{tr}\{\exp(\Psi + A)\}] \leq \text{tr} [\exp\{\Psi + \log(\mathbb{E}[e^A])\}].$$

Inequalities for martingales increments and empirical Gram matrices

Lemma 2.5 (Hoeffding inequality for norm-subGaussian martingale increments). *Let the d -dimensional random vectors $Z_1, \dots, Z_n$ and the natural filtration $\mathcal{F}_n = \sigma(Z_1, \dots, Z_n)$, $\mathcal{F}_0 = \{\Omega, \emptyset\}$, be such that, for all $i = 1, \dots, n$, $\mathbb{E}[Z_i | \mathcal{F}_{i-1}] = 0$ and*

$$\forall t \geq 0, \forall i = 1, \dots, n, \quad \mathbb{P}(\|Z_i\|_2 \geq t | \mathcal{F}_{i-1}) \leq 2 \exp\left(-\frac{t^2}{2\sigma^2}\right) \quad (2.10)$$

for some $\sigma > 0$. Then for any $\delta > 0$, with probability at least $1 - \delta$, we have

$$\left\| \sum_{i=1}^n Z_i \right\|_2 \leq K\sigma\sqrt{n \log(2d/\delta)},$$

where $K = 3$.

Proof. The proof follows from adapting the proof of Lemma 6 in Leluc et al. (2021) working out their Lemma 5 and Corollary 7 from Jin et al. (2019). $\square$

Lemma 2.6. *Define $\varphi_k = \varphi(X_k)$, $Q_k = w_k \varphi_k \varphi_k^\top$, $Y_n = \sum_{k=1}^n Q_k$. Let the constant $L > 0$ be such that $\lambda_{\max}(Q_k) \leq L$ with probability 1. Then, for all $\zeta \in (0, 1)$, we have*

$$\mathbb{P}(\lambda_{\min}(Y_n) \leq (1 - \zeta)n\lambda_{\min}(G)) \leq m \left[\frac{e^{-\zeta}}{(1 - \zeta)^{(1-\zeta)}} \right]^{n\lambda_{\min}(G)/L}.$$

Proof. Let $\mathbb{E}_n$ denote the expectation with respect to $\mathcal{F}_{n-1} = \sigma(X_1, \dots, X_{n-1})$ and define $Z_n = \log(\mathbb{E}_n[e^{\nu Q_n}])$. Using Lemma 2.4 with the measurable w.r.t. $\mathcal{F}_{n-1}$ matrix $\Psi = \nu Y_{n-1}$, we have

$$\begin{aligned} \mathbb{E} [\text{tr}(e^{\nu Y_n})] &= \mathbb{E} [\mathbb{E}_n [\text{tr}(e^{\nu Y_{n-1} + \nu Q_n})]] \leq \mathbb{E} [\text{tr}(e^{\nu Y_{n-1} + \log(\mathbb{E}_n[e^{\nu Q_n}])})] \\ &= \mathbb{E} [\text{tr}(e^{\nu Y_{n-1} + Z_n})]. \end{aligned}$$

Using again Lemma 2.4 with the matrix $\Psi = \nu Y_{n-2} + Z_n$, the last term is upper bounded as

$$\mathbb{E} [\text{tr}(e^{\nu Y_{n-1} + Z_n})] = \mathbb{E} [\mathbb{E}_{n-1} [\text{tr}(e^{\nu Y_{n-2} + \nu Q_{n-1} + Z_n})]] \leq \mathbb{E} [\text{tr}(e^{\nu Y_{n-2} + Z_{n-1} + Z_n})]$$

Applying this inequality several times yields

$$\mathbb{E}[\text{tr}(e^{\nu Y_n})] \leq \mathbb{E}[\text{tr}(e^{\sum_{k=1}^n Z_k})].$$

Applying Lemma 2.3 gives $Z_k \leq \eta(\nu) \mathbb{E}_k[Q_k]$, $\eta(\nu) = L^{-1}(e^{\nu L} - 1)$ for $k = 1, \dots, n$. By Lemma 2.1, we get

$$\mathbb{E}[\text{tr}(e^{\nu Y_n})] \leq \mathbb{E}[\text{tr}(e^{\sum_{k=1}^n Z_k})] \leq \mathbb{E}[\text{tr}(e^{\sum_k \eta(\nu) \mathbb{E}_k[Q_k]})] = \text{tr}(e^{n\eta(\nu)G}).$$

Now applying Lemma 2.2 and taking into account the fact that $\eta(\nu) < 0$ for $\nu < 0$, we have

$$\begin{aligned} \mathbb{P}(\lambda_{\min}(Y_n) \leq t) &\leq \inf_{\nu < 0} e^{-\nu t} \mathbb{E}[\text{tr}(e^{\nu Y_n})] \\ &\leq \inf_{\nu < 0} e^{-\nu t} \text{tr}(e^{n\eta(\nu)G}) \\ &\leq \inf_{\nu < 0} e^{-\nu t} \text{tr}(e^{n\eta(\nu)\lambda_{\min}(G)I_m}) \\ &\leq \inf_{\nu < 0} e^{-\nu t} m e^{n\eta(\nu)\lambda_{\min}(G)}. \end{aligned}$$

We make the change of variables $t = (1 - \zeta)n\lambda_{\min}(G)$ and minimize over $\nu < 0$ the following expression

$$-n\nu(1 - \zeta)\lambda_{\min}(G) + n\eta(\nu)\lambda_{\min}(G).$$

The infimum is attained at $\nu = L^{-1} \log(1 - \zeta)$ with $\eta(\nu) = -\zeta/L$ which gives the inequality of the Lemma. $\square$

Remark 2.4. The term in square brackets in Lemma 2.6 is bounded above by $e^{-\zeta^2/2}$ (Leluc et al. (2021), Lemma 2).

2.7.2 Additional properties of AISCV estimator

Orthogonal projections

Some geometric considerations help to better understand certain properties of the AISCV estimate (2.7). Let $\mathbf{1}_n = (1, \dots, 1)^\top \in \mathbb{R}^n$ be a vector of ones and write

$$f^{(n)} = (f(X_1), \dots, f(X_n))^\top, \quad \Phi = (\varphi_j(X_i))_{\substack{i=1, \dots, n, \\ j=1, \dots, m}}, \quad W = \text{diag}(w_1, \dots, w_n).$$

In matrix form, the weighted least-squares problem (2.7) is

$$(\hat{\alpha}_n, \hat{\beta}_n) \in \arg \min_{(a,b) \in \mathbb{R} \times \mathbb{R}^m} \|W^{1/2}(f^{(n)} - a\mathbf{1}_n - \Phi b)\|_2^2. \quad (2.11)$$

For any function $\psi : \mathbb{R}^d \rightarrow \mathbb{R}^p$, let the operator $P_{n,w}$ return the weighted average of the sequence $\psi(X_1), \dots, \psi(X_n)$ with the weights $w_1, \dots, w_n$, i.e.,

$$P_{n,w}(\psi) = \frac{\sum_{i=1}^n w_i \psi(X_i)}{\sum_{i=1}^n w_i}.$$

The empirically centred integrand and control variates are $f_W^{(n)} = f^{(n)} - \mathbf{1}_n P_{n,w}(f)$ and $\Phi_W = \Phi - \mathbf{1}_n P_{n,w}(\varphi^\top)$. Put $W^{1/2} = \text{diag}(w_1^{1/2}, \dots, w_n^{1/2})$. The solution to (2.11) takes the form

$$\begin{cases} \hat{\alpha}_n = P_{n,w}(f - \hat{\beta}_n^\top \varphi), \\ \hat{\beta}_n \in \arg \min_{b \in \mathbb{R}^m} \|W^{1/2}(f_W^{(n)} - \Phi_W b)\|_2^2, \end{cases} \quad (2.12)$$

If the matrix $\Phi_W^\top W \Phi_W$ is invertible, the optimal vector $\hat{\beta}_n$ is unique and is given by

$$\hat{\beta}_n = (\Phi_W^\top W \Phi_W)^{-1} \Phi_W^\top W f_W^{(n)}. \quad (2.13)$$

Matrix representation

Let us rewrite (2.11) in terms of two nested minimization problems:

$$\hat{\alpha}_n \in \arg \min_{a \in \mathbb{R}} \left[\min_{b \in \mathbb{R}^m} \|W^{1/2}(f^{(n)} - a\mathbf{1}_n - \Phi b)\|_2^2 \right]. \quad (2.14)$$

Let $\Pi \in \mathbb{R}^{n \times n}$ be the orthogonal projection matrix onto the column space of Φ , when $\mathbb{R}^n$ is endowed with the scalar product $\langle x, y \rangle_W = x^\top W y$ for $x, y \in \mathbb{R}^n$. For $v \in \mathbb{R}^n$, we have

$$\Pi v = \Phi \hat{\beta}_n(v) \quad \text{where} \quad \hat{\beta}_n(v) \in \arg \min_{b \in \mathbb{R}^m} \|W^{1/2}(v - \Phi b)\|_2^2.$$

If Φ has rank m , then the solution to the above minimization problem is unique and $\Pi = \Phi(\Phi^\top W \Phi)^{-1} \Phi^\top W$; otherwise, the matrix Π is still uniquely defined, even though there are then multiple solutions $\hat{\beta}_n(v)$. Given $a \in \mathbb{R}$, the minimum in (2.14) over $b \in \mathbb{R}^m$ is attained as soon as $\Phi b = \Pi(f^{(n)} - a\mathbf{1}_n)$. Therefore

$$\hat{\alpha}_n \in \arg \min_{a \in \mathbb{R}} \|W^{1/2}(I_n - \Pi)(f^{(n)} - a\mathbf{1}_n)\|_2^2, \quad (2.15)$$

where I_n is the $n \times n$ identity matrix. Recall the vector e_n in (2.8). In our present notation, we have

$$e_n = (I_n - \Pi)\mathbf{1}_n.$$

Proposition 2.4 (Matrix representation). *The minimizer $\hat{\alpha}_n$ in (2.15) is unique if and only if $e_n \neq 0$, in which case the normalized AISCV estimate is*

$$I_n^{(\text{aiscv})}(f) = \hat{\alpha}_n = \frac{\mathbf{1}_n^\top (I_n - \Pi)^\top W (I_n - \Pi) f^{(n)}}{\mathbf{1}_n^\top (I_n - \Pi)^\top W (I_n - \Pi) \mathbf{1}_n} = \frac{\mathbf{1}_n^\top (I_n - \Pi)^\top W f^{(n)}}{\mathbf{1}_n^\top (I_n - \Pi)^\top W \mathbf{1}_n}. \quad (2.16)$$

Proof. The objective function for which the expression on the right-hand side of (2.16) is given by

$$a^2 \mathbf{1}_n^\top (I_n - \Pi)^\top W (I_n - \Pi) \mathbf{1}_n - 2a \mathbf{1}_n^\top (I_n - \Pi)^\top W (I_n - \Pi) f^{(n)} + \text{constant},$$

where the unspecified constant does not depend on a . The coefficient of a^2 is equal to $e_n^\top W e_n$, which is positive if and only if $e_n \neq 0$. The latter is thus a necessary and sufficient for the minimizer $\hat{\alpha}_n$ to exist and be unique. In that case, the objective function is a convex quadratic function in a , whose minimizer is easily seen to be equal to the stated expression. $\square$

2.7.3 Proofs of the main results

Proof of Proposition 2.1

Proof. We start from Proposition 2.4. Recall that $e_n = (I_n - \Pi) \mathbf{1}_n$. Since $\Pi^\top W = W \Pi$ and $\Pi^2 = \Pi$, we find $(I_n - \Pi)^\top W (I_n - \Pi) = (I_n - \Pi)^\top W$. We obtain

$$\mathbf{1}_n^\top (I_n - \Pi)^\top W (I_n - \Pi) f^{(n)} = \mathbf{1}_n^\top (I_n - \Pi)^\top W f^{(n)} = e_n^\top W f^{(n)} = \sum_{i=1}^n w_i e_{n,i} f(X_i),$$

and similarly $\mathbf{1}_n^\top (I_n - \Pi)^\top W (I_n - \Pi) f^{(n)} = \sum_{i=1}^n w_i e_{n,i}$. $\square$

Proof of Proposition 2.2

Proof. If $f = \alpha + \beta^\top \varphi$ for some $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}^m$, then the minimum in (2.7) is clearly attained for $\hat{\alpha}_n = \alpha$ and $\hat{\beta}_n = \beta$. $\square$

Proof of Proposition 2.3

Proof. In (2.7), if b ranges over $\mathbb{R}^m$, then $A^\top b$ ranges over $\mathbb{R}^m$ too, since A is invertible. It follows that the solutions $\hat{\alpha}_n$ in (2.7) do not change if we replace φ by $A\varphi$, since $b^\top A\varphi = (A^\top b)^\top \varphi$. $\square$

Proof of Theorem 2.1

Proof.

Step 1. Working out the probability of several bounds. In Step 1, we gather several elementary bounds that will be useful to establish more advanced bounds in Step 2.

Bound 1. To control $|\sum_{i=1}^n w_i \varepsilon(X_i)|$, we apply Lemma 2.5 with Z_i equal to $w_i \varepsilon(X_i)$. We have $\mathbb{E}[w_i \varepsilon(X_i) | \mathcal{F}_{i-1}] = 0$ and by Assumption 3,

$$\mathbb{P}[|w_i \varepsilon(X_i)| > t | \mathcal{F}_{i-1}] \leq 2 \exp(-t^2/(2\tau^2))$$

holds, and the sub-Gaussian variance factor is simply τ^2 . Therefore, with probability at least $1 - \delta/5$, we have

$$\left| \sum_{i=1}^n w_i \varepsilon(X_i) \right| \leq K\tau \sqrt{n \log(10/\delta)}.$$

Bound 2. For the term $\|\sum_{i=1}^n w_i \bar{\varphi}(X_i)\|_2$, we apply Lemma 2.5 with Z_i equal to $w_i \bar{\varphi}(X_i)$. By Assumptions 2 and 1, we have $\|w_i \bar{\varphi}(X_i)\|_2 \leq c \|\bar{\varphi}(X_i)\|_2 \leq c\sqrt{B}$, which implies that $w_i \bar{\varphi}(X_i)$ is sub-Gaussian (conditionally on $\mathcal{F}_{i-1}$) with variance factor $c^2 B$ (Boucheron et al., 2013, Lemma 2.2). Hence (2.10) is satisfied with $\sigma^2 = c^2 B$. Thus, with probability at least $1 - \delta/5$, the inequality

$$\left\| \sum_{i=1}^n w_i \bar{\varphi}(X_i) \right\|_2 \leq Kc \sqrt{nB \log(10m/\delta)}$$

holds.

Bound 3. Now we treat the term $\|\sum_{i=1}^n w_i \bar{\varphi}(X_i) \varepsilon(X_i)\|_2$ applying again Lemma 2.5 but this time with Z_i equal to $w_i \bar{\varphi}(X_i) \varepsilon(X_i)$. We have that $\|w_i \bar{\varphi}(X_i) \varepsilon(X_i)\|_2 \leq \sqrt{B} |w_i \varepsilon(X_i)|$. By Assumption 3, we have, for all $t > 0$,

$$\begin{aligned} \mathbb{P}[\|w_i \bar{\varphi}(X_i) \varepsilon(X_i)\|_2 > t | \mathcal{F}_i] &\leq \mathbb{P}[\sqrt{B} |w_i \varepsilon(X_i)| > t | \mathcal{F}_i] \\ &\leq 2 \exp\left(-\frac{t^2}{2B\tau^2}\right), \end{aligned}$$

and (2.10) holds with $\sigma^2 = B\tau^2$. Lemma 2.5 then implies that, with probability at least $1 - \delta/5$,

$$\left\| \sum_{i=1}^n w_i \bar{\varphi}(X_i) \varepsilon(X_i) \right\|_2 \leq K \sqrt{nB\tau^2 \log(10m/\delta)}.$$

Bound 4. By Lemma 2.6 and Remark 2.4, we have, with probability at least $1 - \delta/5$,

$$\lambda_{\min} \left(\sum_{i=1}^n w_i \bar{\varphi}(X_i) \bar{\varphi}^\top(X_i) \right) > (1 - \zeta) n \lambda_{\min}(G) = (1 - \zeta) n,$$

where, by Assumption 2, $G = \int f \bar{\varphi} \bar{\varphi}^\top d\lambda = I$, ζ satisfies the equation

$$m \exp\left(\frac{-\zeta^2 n}{2L}\right) = \delta/5.$$

with $L = cB$ according to Assumptions 2 and 1. Solving the last equation, we obtain

$$\zeta = \sqrt{\frac{2L \log(5m/\delta)}{n}}.$$

We choose $\zeta \leq 1/2$ which gives the condition $n \geq 8cB \log(5m/\delta)$ and, with probability at least $1 - \delta/5$,

$$\lambda_{\min} \left(\sum_{i=1}^n w_i \bar{\varphi}(X_i) \bar{\varphi}^\top(X_i) \right) > (1 - \zeta) n \geq n/2. \quad (2.17)$$

Bound 5. Now we consider the term $\sum_{i=1}^n w_i$. Since $-1 \leq w_i - 1 \leq c$, $|w_i - 1|$ is bounded by c , and $w_i - 1$ is sub-Gaussian with variance factor c^2 . This makes the inequality required in Lemma 2.5 valid and henceforth

$$\left| \sum_{i=1}^n (w_i - 1) \right| \leq Kc \sqrt{n \log(10/\delta)}$$

or

$$-Kc \sqrt{n \log(10/\delta)} + n \leq \sum_{i=1}^n w_i \leq Kc \sqrt{n \log(10/\delta)} + n.$$

We want to have $Kc \sqrt{n \log(10/\delta)} \leq n/2$. It holds if $\sqrt{n} \geq 2Kc \sqrt{\log(10/\delta)}$. Then we get that $n/2 = n - n/2 \leq n - Kc \sqrt{n \log(10/\delta)} \leq \sum_{i=1}^n w_i$. Therefore, with probability at least $1 - \delta/5$, it holds that

$$\sum_{i=1}^n w_i \geq n/2.$$

Step 2. Extending the previous elementary bounds on appropriate quantities. The work in this step consists in showing that under the five previous bounds, and therefore with probability at least $1 - \delta$, we have that

$$\lambda_{\min} \left(\sum_{i=1}^n w_i \bar{\varphi}_W(X_i) \bar{\varphi}_W(X_i)^\top \right) \geq n/4, \quad (2.18)$$

$$\left\| \sum_{i=1}^n w_i \bar{\varphi}_W(X_i) \varepsilon_W(X_i) \right\|_2 \leq 2K\tau \sqrt{nB \log(10m/\delta)}. \quad (2.19)$$

We start by proving (2.18). Recognizing a covariance, we get

$$P_{n,w}\{\bar{\varphi}_W \bar{\varphi}_W^\top\} = P_{n,w}(\bar{\varphi} \bar{\varphi}^\top) - P_{n,w}(\bar{\varphi})P_{n,w}(\bar{\varphi})^\top,$$

and then, using the Cauchy-Schwarz inequality, we have

$$\lambda_{\min}(P_{n,w}\{\bar{\varphi}_W \bar{\varphi}_W^\top\}) \geq \lambda_{\min}(P_{n,w}(\bar{\varphi} \bar{\varphi}^\top)) - \|P_{n,w}(\bar{\varphi})\|_2^2$$

or, equivalently,

$$\begin{aligned} \lambda_{\min}\left(\sum_{i=1}^n w_i \bar{\varphi}_W(X_i) \bar{\varphi}_W(X_i)^\top\right) &\geq \lambda_{\min}\left(\sum_{i=1}^n w_i \bar{\varphi}(X_i) \bar{\varphi}(X_i)^\top\right) \\ &\quad - \left\|\sum_{i=1}^n w_i \bar{\varphi}(X_i)\right\|_2^2 / \sum_{i=1}^n w_i. \end{aligned}$$

From Bound 2 and Bound 5,

$$\left\|\sum_{i=1}^n w_i \bar{\varphi}(X_i)\right\|_2^2 / \sum_{i=1}^n w_i \leq \frac{K^2 c^2 B n \log(10m/\delta)}{n/2} = 2K^2 c^2 B \log(10m/\delta)$$

Using Bound 4 and the previous inequality, it follows that

$$\lambda_{\min}\left(\sum_{i=1}^n w_i \bar{\varphi}_W(X_i) \bar{\varphi}_W(X_i)^\top\right) \geq n/2 - 2K^2 c^2 B \log(10m/\delta).$$

If $n \geq 8K^2 c^2 B \log(10m/\delta)$,

$$\lambda_{\min}\left(\sum_{i=1}^n w_i \bar{\varphi}_W(X_i) \bar{\varphi}_W(X_i)^\top\right) \geq n/4.$$

We have just obtained (2.18).

Let us now establish (2.19). Recognizing a covariance, we find

$$P_{n,w}\{\bar{\varphi}_W \varepsilon_W\} = P_{n,w}(\bar{\varphi} \varepsilon) - P_{n,w}(\bar{\varphi})P_{n,w}(\varepsilon),$$

and it follows that

$$\|P_{n,w}\{\bar{\varphi}_W \varepsilon_W\}\|_2 \leq \|P_{n,w}(\bar{\varphi} \varepsilon)\|_2 + \|P_{n,w}(\bar{\varphi})\|_2 \|P_{n,w}(\varepsilon)\|_2,$$

or, equivalently,

$$\left\|\sum_{i=1}^n w_i \bar{\varphi}_W(X_i) \varepsilon_W(X_i)\right\|_2 \leq \left\|\sum_{i=1}^n w_i \bar{\varphi}(X_i) \varepsilon(X_i)\right\|_2 + \|P_{n,w}(\bar{\varphi})\|_2 \left|\sum_{i=1}^n w_i \varepsilon(X_i)\right|.$$

Now using Bound 2 and 5, we find

$$\|P_{n,w}(\bar{\varphi})\|_2 \leq 2Kc\sqrt{\frac{B\log(10m/\delta)}{n}}, \quad (2.20)$$

which combined with Bound 1 leads to

$$\begin{aligned} \|P_{n,w}(\bar{\varphi})\|_2 \left| \sum_{i=1}^n w_i \varepsilon(X_i) \right| &\leq 2K^2 c \tau \sqrt{B \log(10m/\delta) \log(10/\delta)} \\ &\leq 2K^2 c \tau \sqrt{B} \log(10m/\delta). \end{aligned}$$

The previous inequality and Bound 3 gives

$$\begin{aligned} \left\| \sum_{i=1}^n w_i \bar{\varphi}_W(X_i) \varepsilon_W(X_i) \right\|_2 &\leq K \tau \sqrt{nB \log(10m/\delta)} + 2K^2 c \tau \sqrt{B} \log(10m/\delta) \\ &= K \tau \sqrt{nB \log(10m/\delta)} \left(1 + 2Kc \sqrt{\frac{\log(10m/\delta)}{n}} \right) \\ &\leq 2K \tau \sqrt{nB \log(10m/\delta)} \end{aligned}$$

if $n \geq 4K^2 c^2 \log(10m/\delta)$.

The condition $n \geq 8K^2 c^2 B \log(10m/\delta)$ (used in establishing (2.18)) implies $n \geq 4K^2 c^2 \log(10m/\delta)$ (used in proving (2.19)), $n \geq 8cB \log(5m/\delta)$ (used in Bound 4) and $n \geq 4K^2 c^2 \log(10/\delta)$ (used in Bound 5) since $m \geq 1$, $B \geq m$ and $c \geq 1$. Therefore, the constant C_1 from the statement of the theorem equals $8K^2$.

Step 3. End of the proof. The quantity to be bounded can be written as a sum of two terms as follows

$$I_n^{(\text{aiscv})}(f, \hat{\beta}_n) - \int_{\mathbb{R}^d} f(x) p(x) dx = P_{n,w}\{\varepsilon\} + P_{n,w}\{\varphi\}^\top (\beta^* - \hat{\beta}_n).$$

Using Bounds 1 and 5, the first term in the right-hand side satisfies

$$|P_{n,w}\{\varepsilon\}| \leq 2K \tau \sqrt{\frac{\log(10/\delta)}{n}}.$$

This corresponds to the first term in the bound of the theorem where the constant C_2 equals $2K$. Hence, it remains to show that

$$|P_{n,w}\{\varphi\}^\top (\beta^* - \hat{\beta}_n)| \leq C_3 c B \tau \log(10m/\delta)/n.$$

Introducing $G^{-1/2}G^{1/2}$, we obtain

$$P_{n,w}\{\varphi\}^\top (\beta^* - \hat{\beta}_n) = P_{n,w}\{\bar{\varphi}\}^\top G^{1/2}(\beta^* - \hat{\beta}_n).$$

Then, using the identity

$$(\hat{\beta}_n - \beta^*) = (\Phi_W^\top W \Phi_W)^{-1} \Phi_W^\top W \varepsilon_W^{(n)}$$

and the Cauchy-Schwarz inequality yields

$$\begin{aligned} \left| P_{n,w}\{\varphi\}^\top (\beta^* - \hat{\beta}_n) \right| &\leq \|P_{n,w}\{\bar{\varphi}\}\|_2 \|G^{1/2}(\beta^* - \hat{\beta}_n)\|_2 \\ &\leq \|P_{n,w}\{\bar{\varphi}\}\|_2 \left\| G^{1/2}(\Phi_W^\top W \Phi_W)^{-1} \Phi_W^\top W \varepsilon_W^{(n)} \right\|_2 \\ &\leq \|P_{n,w}\{\bar{\varphi}\}\|_2 \left\| G^{1/2}(\Phi_W^\top W \Phi_W)^{-1} G^{1/2} \right\|_2 \\ &\quad \times \left\| G^{-1/2} \Phi_W^\top W \varepsilon_W^{(n)} \right\|_2. \end{aligned}$$

By (2.18), we have

$$\begin{aligned} \left\| G^{1/2}(\Phi_W^\top W \Phi_W)^{-1} G^{1/2} \right\|_2 &= \left\| \left(\sum_{i=1}^n w_i \bar{\varphi}_W(X_i) \bar{\varphi}_W(X_i)^\top \right)^{-1} \right\|_2 \\ &= \left[\lambda_{\min} \left(\sum_{i=1}^n w_i \bar{\varphi}_W(X_i) \bar{\varphi}_W(X_i)^\top \right) \right]^{-1} \leq 4/n. \end{aligned}$$

From (2.19) and (2.20), it follows that

$$\begin{aligned} \left| P_{n,w}\{\varphi\}^\top (\beta^* - \hat{\beta}_n) \right| &\leq 2K \sqrt{\frac{B \log(10m/\delta)}{n}} \frac{8Kc\tau \sqrt{nB \log(10m/\delta)}}{n} \\ &= 16K^2 cB\tau \frac{\log(10m/\delta)}{n}. \end{aligned}$$

Therefore, the constant C_3 from the statement of the theorem equals $16K^2$. $\square$

2.7.4 Additional numerical results

Parameters. In all simulations, the sampling policy is taken within the family of multivariate Student t distributions of degree ν denoted by $\{q_{\mu, \Sigma_0} : \mu \in \mathbb{R}^d\}$ with $\Sigma_0 = \sigma_0 I_d(\nu-2)/\nu$ and $\nu > 2, \sigma_0 > 0$. Similarly to Portier and Delyon (2018), the mean μ_t is updated at each stage $t = 1, \dots, T$ by the generalized method of moments (GMM), leading to

$$\mu_t = \frac{\sum_{s=1}^t \sum_{i=1}^{n_s} w_{s,i} X_{s,i}}{\sum_{s=1}^t \sum_{i=1}^{n_s} w_{s,i}}.$$

The allocation policy is fixed to $n_t = 1000$ and the number of stages is $T \in \{5; 10; 20; 30; 50\}$. The different Monte Carlo estimates are compared by their mean squared error (MSE) obtained over 100 independent replications. In other words, for each method that returns $\hat{I}(f)$, the mean square error is computed as the average of $\|\hat{I}(f) - I(f)\|_2^2$ computed over 100 replicates of $\hat{I}(f)$. When the integrand is real-valued, this quantity is scaled as $([\hat{I}(f) - I(f)]/I(f))^2$.

The experiments were performed on a laptop Intel Core i7-10510U CPU $1.80\text{GHz} \times 8$.

Synthetic examples: integration on $[0, 1]^d$

We seek to integrate functions f with respect to the uniform density $p(x) = 1$ for $x \in [0, 1]^d$ in dimensions $d \in \{4; 8\}$. We rely on Legendre polynomials for the control variates. Consider the integrands $f_1(x) = 1 + \sin(\pi(2d^{-1} \sum_{i=1}^d x_i - 1))$, $f_2(x) = \prod_{i=1}^d (2/\pi)^{1/2} x_i^{-1} e^{-\log(x_i)^2/2}$ and $f_3(x) = \prod_{i=1}^d \log(2) 2^{1-x_i}$, all of which integrate to 1 on $[0, 1]^d$.

None of the integrands is a linear combination of the control variates. The policy parameters are $\mu_0 = (0.5, \dots, 0.5) \in \mathbb{R}^d$, $v = 8$, and $\sigma_0 = 0.1$. The control variates are built out of tensor products of Legendre polynomials where the degree ℓ_j equals 0 for all but two coordinates, leading to a total number of $m = kd + k^2d(d-1)/2$ control variates. The maximum degree in each variable is $k = 6$, yielding $m = 240$ and $m = 1056$ control variates in dimensions $d = 4$ and $d = 8$ respectively. Figure 2.1 presents the boxplots of the AIS and AISCV estimates.

Figure 2.4 presents the boxplots of the different estimates and Table 2.1 gathers the numerical values of the mean squared errors. As a natural competitor to our AISCV estimator, we also implemented the weighted version of standard AIS called *w-AIS* introduced in Portier and Delyon (2018). Interestingly, such a method presents similar or even worse performance than the standard AIS estimate for dimension $d = 4$ but better results for dimension $d = 8$. This good behavior is illustrated in Figure 2.4b and Figure 2.4d. Accordingly, the values of the MSE for *w-AIS* are smaller than the one of AIS in dimension $d = 8$ but still greater than the ones of AISCV.

Synthetic examples: Gaussian mixtures

General target p and Stein method. In this setting we only assume access to the evaluations of the target density p . We consider the classical example introduced in Cappé et al. (2008) where p is a mixture of two Gaussian distri-

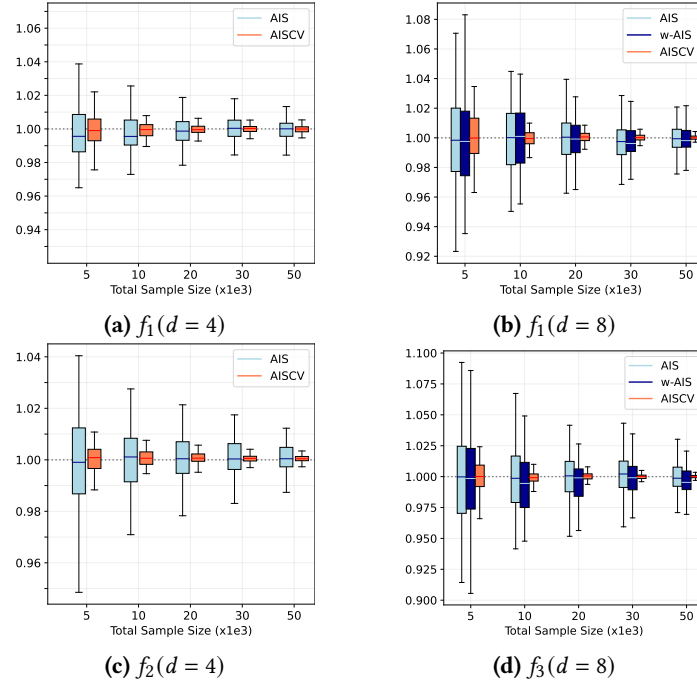


Figure 2.4: Integration on $[0, 1]^d$: boxplots of estimates $I_n^{(\text{ais})}(f)$ and $I_n^{(\text{aiscv})}(f)$ with integrands f_1, f_2, f_3 in dimensions $d \in \{4; 8\}$ obtained over 100 replications. The true integral equals 1.

Sample Size n		5,000	10,000	20,000	30,000	50,000
Integrand	Method					
f_1 ($d=4$)	AIS	$2.9e-4$	$1.5e-4$	$7.8e-5$	$5.8e-5$	$3.7e-5$
	wAIS	$3.0e-4$	$1.6e-4$	$8.3e-5$	$6.5e-5$	$4.1e-5$
	AISCV	$9.7e-5$	$1.9e-5$	$1.0e-5$	$7.5e-6$	$4.3e-6$
f_1 ($d=8$)	AIS	$8.7e-4$	$4.6e-4$	$2.3e-4$	$1.9e-4$	$1.0e-4$
	wAIS	$9.2e-4$	$4.6e-4$	$2.2e-4$	$1.6e-4$	$9.0e-5$
	AISCV	$3.2e-4$	$3.2e-5$	$1.1e-5$	$6.0e-6$	$2.5e-6$
f_2 ($d=4$)	AIS	$3.4e-4$	$1.3e-4$	$7.6e-5$	$5.9e-5$	$3.1e-5$
	wAIS	$3.7e-4$	$1.6e-4$	$1.2e-4$	$1.1e-4$	$7.9e-5$
	AISCV	$3.1e-5$	$1.0e-5$	$4.9e-6$	$2.6e-6$	$1.5e-6$
f_3 ($d=8$)	AIS	$1.6e-3$	$7.8e-4$	$4.0e-4$	$3.3e-4$	$1.9e-4$
	wAIS	$1.5e-3$	$7.3e-4$	$3.6e-4$	$2.7e-4$	$1.5e-4$
	AISCV	$1.7e-4$	$2.1e-5$	$7.8e-6$	$4.3e-6$	$1.8e-6$

Table 2.1: Mean Square Errors for f_1, f_2, f_3 with AIS, wAIS (Portier and Delyon, 2018) and AISCV in dimensions $d \in \{4; 8\}$ obtained over 100 replications.

butions. The control variates are built using Stein's method with polynomial maps of degree $Q \in \{2; 3\}$ leading to a number of control variates $m \in \{14; 34\}$ in dimension $d = 4$ and $m \in \{44; 164\}$ in dimension $d = 8$ respectively.

Isotropic case. Let $p_\Sigma(x) = 0.5\Phi_\Sigma(x - \mu) + 0.5\Phi_\Sigma(x + \mu)$ where $\Sigma = I_d/d$, $\mu = (1, \dots, 1)^\top / (2\sqrt{d})$ and Φ_Σ is the multivariate normal density function with zero mean and covariance matrix Σ . Note that the Euclidean distance between the two mixture centers is independent of the dimension as it equals 1. The initial density q_0 is the multivariate student distribution with mean $(1, -1, 0, \dots, 0)^\top / \sqrt{d}$ and variance $(5/d)I_d$. The initial mean value differs from the null vector to prevent the naive algorithm using the initial density from having good results (due to the symmetry).

Anisotropic case. In this case, the mixture is unbalanced and each Gaussian is anisotropic. The target density is $p_V(x) = 0.75\Phi_V(x - \mu) + 0.25\Phi_V(x + \mu)$ where $\mu = (1, \dots, 1)^\top / (2\sqrt{d})$ and $V = \text{diag}(10, 1, \dots, 1)/d$. The initial density q_0 is the same as for the isotropic case.

Figure 2.5 presents the boxplots of the mean square error $\|\hat{I}(f) - I(f)\|_2^2$ and Table 2.2 gathers the associated numerical values.

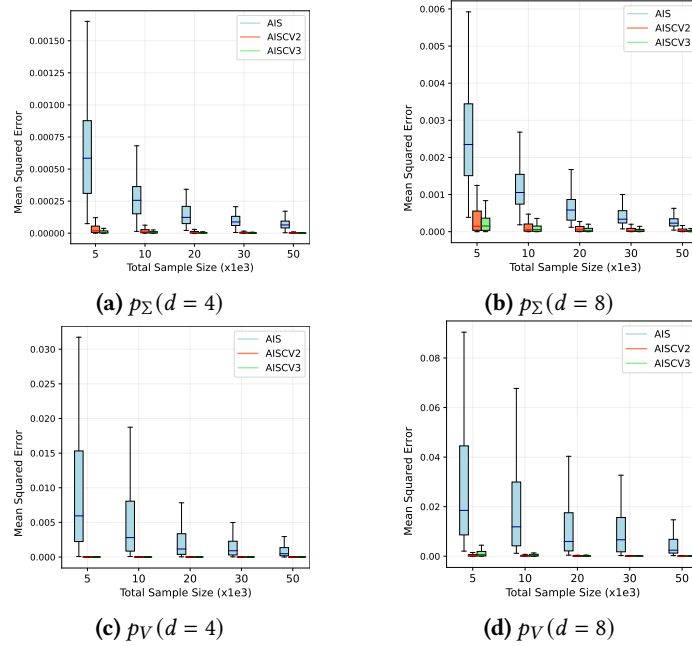


Figure 2.5: Boxplots for $\|\hat{I}(f) - I(f)\|_2^2$ for $f(x) = x$ with target isotropic p_Σ and anisotropic p_V in dimensions $d \in \{4; 8\}$ obtained over 100 replications.

Sample Target	Size n Method	5,000	10,000	20,000	30,000	50,000
f_Σ ($d = 4$)	AIS	6.9e-4	2.9e-4	1.5e-4	1.1e-4	7.2e-5
	wAIS	6.8e-4	2.9e-4	1.5e-4	1.1e-4	7.3e-5
	AISCV-2	4.1e-5	2.2e-5	9.1e-6	5.6e-6	3.7e-6
	AISCV-3	1.5e-5	8.4e-6	3.7e-6	2.3e-6	1.3e-6
f_Σ ($d = 8$)	AIS	2.7e-3	1.2e-3	6.6e-4	4.1e-4	2.7e-4
	wAIS	2.7e-3	1.2e-3	6.9e-4	4.3e-4	2.8e-4
	AISCV-2	3.7e-4	1.7e-4	1.0e-4	6.8e-5	4.7e-5
	AISCV-3	2.8e-4	1.2e-4	6.3e-5	4.2e-5	2.6e-5
f_V ($d = 4$)	AIS	1.1e-2	5.5e-3	2.2e-3	1.6e-3	9.5e-4
	wAIS	1.1e-2	5.3e-3	2.0e-3	1.3e-3	8.0e-4
	AISCV-2	1.3e-5	7.2e-6	2.9e-6	1.9e-6	1.2e-6
	AISCV-3	1.1e-5	6.6e-6	2.2e-6	1.5e-6	9.6e-7
f_V ($d = 8$)	AIS	4.5e-2	3.2e-2	2.2e-2	1.5e-2	6.8e-3
	wAIS	2.6e-2	1.3e-2	7.8e-3	5.9e-3	3.8e-3
	AISCV-2	4.6e-4	2.8e-4	1.3e-4	9.7e-5	6.0e-5
	AISCV-3	1.4e-3	4.8e-4	1.5e-4	1.1e-4	5.7e-5

Table 2.2: Mean Square Errors $\|\hat{I}(f) - I(f)\|_2^2$ for $f(x) = x$ with target isotropic p_Σ and anisotropic p_V in dimensions $d \in \{4; 8\}$ obtained over 100 replications.

Real-world data: Bayesian linear regression

We place ourselves in the framework of Bayesian linear regression with observations $X \in \mathbb{R}^{N \times d}$ and labels $y \in \mathbb{R}^N$. The posterior distribution $p(\theta|\mathcal{D})$ depends on a Gaussian prior $\pi \sim \mathcal{N}(\mu_a, \Sigma_a)$ and a likelihood function $\ell(\mathcal{D}|\theta) \propto (\sigma^2)^{-N/2} \exp(-(y - X\theta)^\top (y - X\theta)/(2\sigma^2))$ where the noise level is fixed and taken sufficiently large $\sigma = 50$ to account general priors. Observe that the posterior distribution is also Gaussian $\mathcal{N}(\mu_b, \Sigma_b)$ with $\mu_b = \Sigma_b(\sigma^{-2}X^\top y + \Sigma_a^{-1}\mu_a)$ and $\Sigma_b = (\sigma^{-2}X^\top X + \Sigma_a^{-1})^{-1}$. The integrand is $f(\theta) = \|\theta\|_2^2$ and the control variates are built using Stein method described in Section 2.5.1 with degree $Q \in \{1; 2\}$, leading to the AISCV1 and AISCV2 estimators respectively. Observe that when $Q = 2$, the integrand belongs to the linear span of the control variates so the integration should be exact in light of Proposition 2.2.

Datasets and parameters. Some classical datasets from UCI Machine Learning repository Dua and Graff (2019) are considered : *housing* ($N = 506; d = 13; m \in \{12; 104\}$); *abalone* ($N = 4,177; d = 8; m \in \{7; 44\}$); *red wine* ($N = 1,599; d = 11; m \in \{10; 77\}$) and *white wine* ($N = 4,898; d = 11; m \in \{10; 77\}$). The initial density is the multivariate student distribution with $\nu = 10$ degrees of freedom, zero mean and covariance matrix Σ_b .

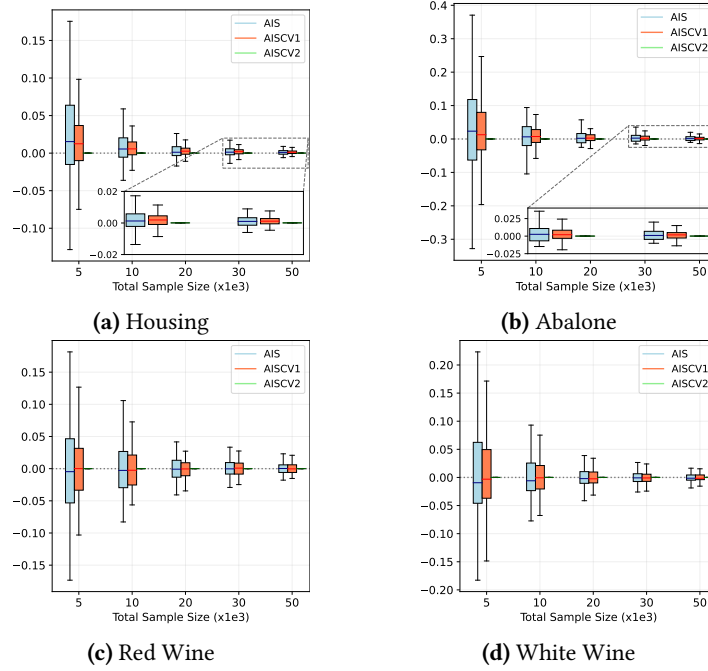


Figure 2.6: Boxplots of $(\hat{I}(f) - I(f))/I(f)$, $f(\theta) = \|\theta\|_2^2$, obtained over 100 replications.

Results. Figure 2.6 presents the boxplots of the error $(\hat{I}(f) - I(f))/I(f)$ and Table 2.3 gathers the associated numerical values. Observe the benefits of using control variates even with polynomials of degree $Q = 1$. Observe that when $Q = 2$, the error of the AISCV2 estimator is almost equal to zero which is in line with Proposition 2.2. Accordingly when looking at the MSE, the AISCV1 error is smaller than the AIS one by a factor ranging between 2 and 10 and the MSE of AISCV2 is of order 10^{-9} .

Monte Carlo Markov Chain. We run a state-of-the-art MCMC method called NUTS (Hoffman et al., 2014), which is a self-tuning variant of Hamiltonian Monte Carlo. It may be hard to compare precisely this method against the AIS based methods since they are different in nature. Indeed, the goal of MCMC methods is to sample from a target distribution whereas AISCV methods are meant for variance reduction. In both cases there are hyperparameters to tune. For AIS-based methods, there is the choice of the policy $(q_i)_{i \geq 0}$, the choice of the control variates and the number of particles n_t to draw at each step. For the NUTS sampler, there is among others, the number of samples used for the tuning phase and the initialization of the Markov kernel. A reasonable comparison is obtained based on the overall number of sampled particles. Table 2.4 above presents the mean squared errors obtained

Dataset	Sample Size n Method	5,000	10,000	20,000	30,000	50,000
Housing	AIS	$2.2e-2$	$4.4e-3$	$3.1e-4$	$2.7e-4$	$2.5e-4$
	AISCV1	$2.9e-3$	$7.0e-4$	$1.7e-4$	$1.6e-4$	$5.2e-5$
	AISCV2	$5.6e-9$	$5.6e-9$	$5.6e-9$	$5.6e-9$	$5.6e-9$
Abalone	AIS	$6.2e-2$	$2.6e-2$	$1.1e-2$	$6.5e-3$	$3.1e-3$
	AISCV1	$6.3e-3$	$1.2e-3$	$4.7e-4$	$3.1e-4$	$1.8e-4$
	AISCV2	$5.1e-9$	$6.1e-9$	$6.1e-9$	$6.1e-9$	$6.1e-9$
Red Wine	AIS	$3.0e-2$	$1.3e-2$	$7.0e-3$	$4.7e-3$	$2.8e-3$
	AISCV1	$3.7e-3$	$1.5e-3$	$8.7e-4$	$6.4e-4$	$4.2e-4$
	AISCV2	$5.1e-10$	$5.1e-10$	$5.1e-10$	$5.1e-10$	$5.1e-10$
White Wine	AIS	$1.1e-2$	$2.6e-3$	$8.1e-4$	$4.2e-4$	$1.8e-4$
	AISCV1	$7.1e-3$	$1.5e-3$	$4.0e-4$	$2.1e-4$	$9.2e-5$
	AISCV2	$2.4e-9$	$2.4e-9$	$2.4e-9$	$2.4e-9$	$2.4e-9$

Table 2.3: Mean Square Errors for different datasets with $f(\theta) = \|\theta\|_2^2$ obtained over 100 replications.

Sample Size n	5,000	10,000	20,000	30,000
Housing	$5.5e-5$	$2.7e-5$	$1.9e-5$	$1.2e-5$
Abalone	$1.8e-4$	$8.6e-5$	$6.7e-5$	$5.6e-5$
Red Wine	$2.7e-4$	$1.8e-4$	$9.5e-5$	$5.2e-5$
White Wine	$3.8e-4$	$1.6e-4$	$8.5e-5$	$7.3e-5$

Table 2.4: MSE with $f(\theta) = \|\theta\|_2^2$ obtained over 30 chains of NUTS sampler.

over 30 chains of NUTS sampler with default configuration of the parameters from the Python library *pymc3* (Patil et al., 2010).

Control Variates and Nearest Neighbors

3

A novel linear integration rule called *control neighbors* is proposed in which nearest neighbor estimates act as control variates to speed up the convergence rate of the Monte Carlo procedure on metric spaces. The main result is the $O(n^{-1/2}n^{-s/d})$ convergence rate – where n stands for the number of evaluations of the integrand and d for the dimension of the domain – of this estimate for Hölder functions with regularity $s \in (0, 1]$, a rate which, in some sense, is optimal. Several numerical experiments validate the complexity bound and highlight the good performance of the proposed estimator. The content of this chapter is based on the paper

R. Leluc, F. Portier, J. Segers and A. Zhuman. (2024).
Speeding up Monte Carlo integration: control neighbors for optimal convergence. *Bernoulli*, in press.

3.1 Introduction

Consider the classical numerical integration problem of approximating the value of an integral $\mu(f) = \int f d\mu$ where μ is a probability measure on a metric space (M, ρ) and the integrand f is a real-valued function on the support of μ . Suppose that random draws from the measure μ are available and calls to the function f are possible. The standard Monte Carlo estimate consists in averaging $f(X_i)$ over $i = 1, \dots, n$, where the particles X_i are drawn independently from μ . For square-integrable integrands, the Monte Carlo estimate has convergence rate $O(n^{-1/2})$ as $n \rightarrow \infty$, whatever the dimension of the domain. In some applications, calls to the integrand may be expensive (Sacks et al., 1989; Doucet et al., 2001) and one may only have access to a small number of evaluations $f(X_i)$, such as in Bayesian inference on complex models (Higdon et al., 2015; Toscano-Palmerin and Frazier, 2022). The $n^{-1/2}$ -rate of the standard Monte Carlo estimate then becomes too slow and leads to highly variable estimates.

As detailed in Novak (2016), the complexity of integration algorithms may be analyzed through the convergence rate of the error. Any randomized pro-

cedure based on n particles yields an estimate $\hat{\mu}_n(f)$ of the integral $\mu(f)$ and the root mean-square error of the procedure is $E[|\hat{\mu}_n(f) - \mu(f)|^2]^{1/2}$. For the specific problem of integration with respect to the uniform measure over the unit cube $[0, 1]^d$, the complexity rate of randomized methods for Lipschitz integrands is known to be $O(n^{-1/2}n^{-1/d})$ (Bakhvalov, 2015; Novak, 2016). Furthermore, when the first s derivatives of the integrand are bounded, the convergence rate becomes $O(n^{-1/2}n^{-s/d})$. These complexity rates are informative as they show that for smooth integrands, the Monte Carlo estimate is suboptimal and leaves room for improvement by relying on the regularity of the integrand. Several approaches are already known for improving upon the Monte Carlo benchmark. They can be classified according to their convergence rates while keeping in mind the lower bound $O(n^{-1/2}n^{-s/d})$.

The control variate method (Rubinstein, 1981; Newton, 1994; Caflisch, 1998; Evans and Swartz, 2000; Glynn and Szechtman, 2002; Glasserman, 2004) is a powerful technique that allows to reduce the variance of the Monte Carlo estimate by approximating the integrand with a function whose integral is known. The Monte Carlo rate can be improved by combining control variates to model f (Oates et al., 2017; Portier and Segers, 2019; Leluc et al., 2021; South et al., 2023). In Portier and Segers (2019), when μ is the uniform distribution on $[-1, 1]^d$ and when using m orthogonal polynomials as control variates, the convergence rate is $O(n^{-1/2}m^{-s/d})$, where s is the regularity of f . However, m is required to be of smaller order than $n^{1/3}$, so that the optimal rate cannot be achieved. By relying on control functions constructed in a reproducing kernel Hilbert space, Oates et al. (2019) obtained the rate $O(n^{-1/2-(a \wedge b)/d+\epsilon})$ for a specific class of integrands where a is related to the smoothness of the target density, b is related to the smoothness of the integrand f and $\epsilon > 0$ hides logarithmic factors and can be arbitrary small. This is done at the expense of computational complexity in $O(n^3)$ due to matrix inversion when solving the underlying kernel ridge regression problem.

Another reliable technique to improve the rate of convergence of standard Monte Carlo is stratification. The sample space is partitioned and particles are sampled from each piece of the partition separately. It has allowed to improve the convergence rate of Monte Carlo estimates (Haber, 1966, 1967) and to derive a general framework called stochastic quadrature rules (Haber, 1969). Recently, Haber's work has been extended to take advantage of higher smoothness in the integrand (Chopin and Gerber, 2022). To the best of our knowledge, the works of Haber (1966) and Chopin and Gerber (2022) are the first ones to provide a feasible method achieving the best rate of convergence for smooth integrands; an algorithm with optimal rate of convergence but difficult to implement is proposed in Krieg and Novak (2017). Some related ideas for building optimal algorithms were already introduced in the seminal

work of Bakhvalov (2015); see also Novak (2016) for a short review and Novak (1988) for more details. A drawback of the methods proposed in Haber (1966) and Chopin and Gerber (2022) is that they are only valid for integration over the unit cube and involve a geometric number ($n = \ell^d$) of evaluations of the integrand f , which may be somewhat restrictive in case the integrand is hard to compute, as in complex Bayesian models.

Interestingly, the stratification method in Chopin and Gerber (2022) relies on a piecewise constant control function with a low bias compared to a traditional regression estimate. Such type of estimator is shown to attain the optimal rate of convergence in case of noiseless evaluations of the function to approximate (Kohler and Krzyżak, 2013; Hinrichs et al., 2020) and this explains the success of their approach to reach to optimal rate of convergence for the integration problem. The same idea of using a low-bias estimate is the starting point of this paper too and is relevant because the integrand f is accessible without noise. Low-bias estimates have also been employed successfully in adaptive rejection sampling (Achddou et al., 2019), allowing to reach the optimal rate. Alternatively, stochastic quadrature rules based on determinantal point processes have been proposed in Bardenet and Hardy (2020) and Coeurjolly et al. (2021). Such methods allow to reduce the root mean squared error to $O(n^{-1/2}n^{-1/(2d)})$ when the integrand f has a continuous derivative. This interesting acceleration still remains slower than the optimal lower bound.

In this paper, a new Monte Carlo method called *control neighbors* is introduced. By using 1-nearest neighbor estimates as control variates, it produces an estimate $\hat{\mu}_n(f)$ of the integral $\mu(f)$ for a probability measure μ on a metric space (M, ρ) such that the measure of a ball of radius $r > 0$ is of the order r^d as $r \rightarrow 0$, uniformly over the space. This novel estimate is shown to achieve the convergence rate $O(n^{-1/2}n^{-s/d})$ for Hölder integrands of regularity $s \in (0, 1]$. The main properties of the control neighbors estimate are as follows:

- (a) It can be obtained under the same framework as standard Monte Carlo, i.e., as soon as one can draw random particles from μ and evaluate the integrand f . In contrast to the classical method of control variates, the existence of control variates with known integrals is not required.
- (b) The method takes the form of a linear integration rule $\sum_{i=1}^n w_{i,n} f(X_i)$ with weights $w_{i,n}$ not depending on the integrand f but only on the particles $X_1, \dots, X_n$. This property is computationally beneficial when several integrals are to be computed with the same measure μ .
- (c) For Lipschitz integrands f , when $s = 1$, the convergence rate is the optimal one $O(n^{-1/2}n^{-1/d})$ (Novak, 2016). Other recent approaches for

general μ (e.g. Oates et al., 2017; Portier and Segers, 2019) do not achieve this rate. Such a rate is achieved by the methods in Haber (1966) and in Chopin and Gerber (2022) provided μ is the uniform distribution on a d -dimensional cube and the number of evaluations of the integrand is geometric, $n = \ell^d$ for some integer $\ell \geq 1$.

- (d) The approach is *post-hoc* in the sense that it can be run after sampling the particles and independently from the sampling mechanism. Although the theory in our paper is restricted to independent random variables, the method can be implemented for other sampling designs including MCMC or adaptive importance sampling.
- (e) The method not only applies to integration with respect to smooth measures on compact subsets of Euclidean space but also with respect to the normalized volume measure on compact Riemannian manifolds such as the unit sphere or the orthogonal group.

A similar approach to the one in this paper (and developed independently) has been proposed in Blanchet et al. (2023) as an example of an algorithm that achieves the complexity bounds for the integration problem; see their Section 4 and in particular their Theorem 4.2. Their study is focused on the derivation of complexity rates and on the existence of methods to achieve these rates. They consider the case where the integrand is evaluated with noise, and this leads to different choices of the number of neighbors (as explained in the proof of their Theorem 4.2). When the noise vanishes, their approach is similar to our 1-nearest neighbor control variate estimate. One major difference remains that, for the proof of their theorem, they need to split the sample into two independent parts, leading to a loss of efficiency in practice. This latter drawback is avoided in this study with the help of a careful analysis of the leave-one-out algorithm.

The outline of the paper is as follows. The mathematical foundations of nearest neighbor estimates are gathered in Section 3.2 with a formal introduction of two different nearest neighbors estimates. The theoretical properties of the control neighbor estimates are stated in Section 3.3. Finally, Section 3.4 reports on several numerical experiments along with some practical remarks on the implementation of the proposed estimates, and Section 3.5 concludes. Section 3.6 contains proofs, auxiliary results and additional experiments.

3.2 From nearest neighbors to control neighbors

This section presents the mathematical framework of nearest neighbor estimates with reminders on Voronoi cells and central quantities for the analysis,

namely the degree of a point and the average cell volume. Next, two *control neighbor* estimates are introduced and some basic properties are stated.

3.2.1 Nearest neighbors

Let $X_1, \dots, X_n$ be independent and identically distributed random variables taking values in a metric space (M, ρ) , drawn from a distribution μ without atoms. Let $x \in M$ denote a generic point.

Definition 3.1 (Nearest neighbors and distances). *The nearest neighbor $\hat{N}_n(x)$ of x among $X_1, \dots, X_n$ and the associated distance $\hat{\tau}_n(x)$ are*

$$\hat{N}_n(x) \in \arg \min_{Y \in \{X_1, \dots, X_n\}} \rho(x, Y), \quad \hat{\tau}_n(x) = \rho(\hat{N}_n(x), x).$$

When the $\arg \min$ is not unique, $\hat{N}_n(x)$ is defined as the one point among the $\arg \min$ having the smallest index. More generally, for $k \in \{1, \dots, n\}$, let $\hat{N}_{n,k}(x)$ denote the k -nearest neighbor of x and $\hat{\tau}_{n,k}(x) = \rho(\hat{N}_{n,k}(x), x)$ the associated distance, breaking ties by the lexicographic order.

The sample $\mathcal{X}_n = \{X_1, \dots, X_n\}$ defines a natural (random) partition of the integration domain when considering the associated Voronoi cells. Any such cell is associated to a given sample point, say X_i , and contains all the points x of which the nearest neighbor is X_i .

Definition 3.2 (Voronoi cells and volumes). *The Voronoi cells of $X_1, \dots, X_n$ are*

$$\forall i = 1, \dots, n, \quad S_{n,i} = \{x \in M : \hat{N}_n(x) = X_i\},$$

with Voronoi volumes $V_{n,i} = \mu(S_{n,i})$.

The 1-NN estimate of f is defined as $\hat{f}_n(x) = f(\hat{N}_n(x))$ for all $x \in \mathbb{R}^d$ and is piece-wise constant on the Voronoi cells, i.e., $\hat{f}_n(x) = \sum_{i=1}^n f(X_i) \mathbb{1}_{S_{n,i}}(x)$.

The leave-one-out rule is a general technique to introduce independence between the prediction and the evaluation points. It is used as a cross-validation strategy in order to tune hyper-parameters of statistical procedures (Stone, 1974; Craven and Wahba, 1978). The leave-one-out version of $\hat{f}_n$ without X_i is denoted by $\hat{f}_n^{(i)}$ and is obtained in the exact same way as $\hat{f}_n$ except that a slightly different sample—in which the i -th observation has been removed—is used.

Definition 3.3 (Leave-one-out neighbors, Voronoi cells and volumes). *Let $i \in \{1, \dots, n\}$ and $\mathcal{X}_n^{(i)} = \{X_1, \dots, X_n\} \setminus \{X_i\}$. The leave-one-out neighbor of*

$x \in M$ is

$$\hat{N}_n^{(i)}(x) \in \arg \min_{Y \in \mathcal{X}_n^{(i)}} \rho(x, Y).$$

When the above $\arg \min$ is not unique, $\hat{N}_n^{(i)}(x)$ is defined as the one point among the $\arg \min$ having the smallest index. The leave-one-out Voronoi cell $S_{n,j}^{(i)}$ denotes the j -th Voronoi cell in $\mathcal{X}_n^{(i)}$, i.e.,

$$\forall j \in \{1, \dots, n\} \setminus \{i\}, \quad S_{n,j}^{(i)} = \left\{ x \in M : \hat{N}_n^{(i)}(x) = X_j \right\}.$$

The leave-one-out Voronoi volume is defined as $V_{n,j}^{(i)} = \mu(S_{n,j}^{(i)})$.

The leave-one-out 1-NN predictor used in Section 3.2.2 to define the proposed integral estimate is $\hat{f}_n^{(i)}(x) = f(\hat{N}_n^{(i)}(x))$. A key property is that $\hat{f}_n^{(i)}$ and $\hat{f}_n$ coincide on $S_{n,j}$ for $j \neq i$. On the cell $S_{n,i}$, when the function f is Hölder with regularity $s \in (0, 1]$, the supremum distance between $\hat{f}_n^{(i)}$ and $\hat{f}_n$ is of the same order as the nearest neighbor distance to the power s . In terms of the $L^1(\mu)$ -norm, their difference is even smaller if the cell $S_{n,i}$ has a small μ -volume. Relevant for our numerical integration problem is that the average of the integrals $\mu(\hat{f}_n^{(i)})$ is close to $\mu(\hat{f}_n)$.

Lemma 3.1. *In terms of $\bar{f}_n(x) = \sum_{i=1}^n \hat{f}_n^{(i)}(x) \mathbb{1}_{S_{n,i}}(x)$, we have $\sum_{i=1}^n (\mu(\hat{f}_n^{(i)}) - \mu(\hat{f}_n)) = \mu(\bar{f}_n - \hat{f}_n)$.*

Enumerating how often a point X_j is the nearest neighbor of points X_i for $i \neq j$ reflects how much X_j is surrounded within the sample. Another quantification of the isolation of a point X_j is obtained by summing the Voronoi volumes $V_{n,j}^{(i)}$ over $i \neq j$. These two notions are formally stated in the next definition.

Definition 3.4 (Degree and cumulative volume). *For all $j = 1, \dots, n$, the degree $\hat{d}_{n,j}$ represents the number of times X_j is a nearest neighbor of a point X_i for $i \neq j$. The associated j -th cumulative Voronoi volume is denoted by $\hat{c}_{n,j}$, that is,*

$$\hat{d}_{n,j} = \sum_{i:i \neq j} \mathbb{1}_{S_{n,j}^{(i)}}(X_i) \quad \text{and} \quad \hat{c}_{n,j} = \sum_{i:i \neq j} V_{n,j}^{(i)}.$$

Interestingly, the degree of a point and its cumulative Voronoi volume have the same expectation: $E[\hat{d}_{n,j}] = E[\hat{c}_{n,j}] = 1$. The two quantities $\hat{d}_{n,j}$ and $\hat{c}_{n,j}$ will be useful in Proposition 3.1 below to express the control neighbors estimate as a linear integration rule. For now, we note that weighted sums of $f(X_j)$ using $\hat{d}_{n,j}$ and $\hat{c}_{n,j}$ as weights are related to the leave-one-out estimate.

Lemma 3.2. *It holds that*

$$\sum_{i=1}^n f(X_i) \hat{d}_{n,i} = \sum_{i=1}^n \hat{f}_n^{(i)}(X_i) \quad \text{and} \quad \sum_{i=1}^n f(X_i) \hat{c}_{n,i} = \sum_{i=1}^n \mu(\hat{f}_n^{(i)}).$$

3.2.2 Control neighbors

With the help of previous notation, we now introduce the two *control neighbor* estimates

$$\hat{\mu}_n^{(\text{NN})}(f) = \frac{1}{n} \sum_{i=1}^n \left[f(X_i) - \left\{ \hat{f}_n^{(i)}(X_i) - \mu(\hat{f}_n) \right\} \right] \quad (3.1)$$

$$\hat{\mu}_n^{(\text{NN-loo})}(f) = \frac{1}{n} \sum_{i=1}^n \left[f(X_i) - \left\{ \hat{f}_n^{(i)}(X_i) - \mu(\hat{f}_n^{(i)}) \right\} \right]. \quad (3.2)$$

The first one, $\hat{\mu}_n^{(\text{NN})}(f)$, is the output in the nearest neighbor algorithm in Section 3.4.1. The second one, $\hat{\mu}_n^{(\text{NN-loo})}(f)$, is a slight modification of $\hat{\mu}_n^{(\text{NN})}(f)$ that is an unbiased estimate of $\mu(f)$. Indeed, a simple conditioning argument implies

$$\mathbb{E} \left[\hat{f}_n^{(i)}(X_i) - \mu(\hat{f}_n^{(i)}) \right] = \mathbb{E} \left[\mathbb{E} \left[\hat{f}_n^{(i)}(X_i) - \mu(\hat{f}_n^{(i)}) \mid \mathcal{X}_n^{(i)} \right] \right] = 0,$$

which is sufficient to get the zero-bias property

$$\mathbb{E} \left[\hat{\mu}_n^{(\text{NN-loo})}(f) \right] = \mu(f).$$

The estimate $\hat{\mu}_n^{(\text{NN})}$ is not unbiased, but under the conditions of Theorem 3.1 below, its bias is of smaller order than its root mean squared error; see Remark 3.4 below. Moreover, since $\hat{f}_n^{(i)}$ is similar to the 1-NN estimate $\hat{f}_n$ of f based on the full sample $X_1, \dots, X_n$, their integrals should be close. This intuition is confirmed in Proposition 3.3 in the Appendix. Consequently, the estimate (3.2) will play an important role in the theory. However, computing the terms $\mu(\hat{f}_n^{(i)})$ for $i = 1, \dots, n$ requires the evaluation of n additional integrals. In practice, the working estimate is (3.1) as it involves fewer computations. Further, it is worthwhile to note that our two control neighbor estimates are *not* based on a single approximation of the integrand f but rather on n different approximations $\hat{f}_n^{(i)}$.

The two control neighbor estimates in (3.1) and (3.2) can be expressed as linear integration rules of the form $\sum_{i=1}^n w_{i,n} f(X_i)$ with weights $w_{i,n}$ not depending on the integrand f . The weights involve the degrees $\hat{d}_{n,i}$ and the (cumulative) volumes $V_{n,i}$ and $\hat{c}_{n,i}$ in Definitions 3.2 and 3.4.

Proposition 3.1 (Quadrature rules). *The estimates $\hat{\mu}_n^{(\text{NN})}(f)$ and $\hat{\mu}_n^{(\text{NN-loo})}(f)$ can be expressed as linear estimates of the form*

$$\hat{\mu}_n^{(\text{NN-loo})}(f) = \sum_{i=1}^n w_{i,n}^{(\text{NN-loo})} f(X_i) \quad \text{and} \quad \hat{\mu}_n^{(\text{NN})}(f) = \sum_{i=1}^n w_{i,n}^{(\text{NN})} f(X_i)$$

where $w_{i,n}^{(\text{NN-loo})} = (1 + \hat{c}_{n,i} - \hat{d}_{n,i})/n$ and $w_{i,n}^{(\text{NN})} = (1 + nV_{n,i} - \hat{d}_{n,i})/n$.

The weights of the two estimates satisfy $\sum_{i=1}^n w_{i,n} = 1$, meaning that the integration rules are exact for constant functions. In the light of Proposition 3.1, the proposed estimate $\hat{\mu}_n^{(\text{NN})}(f)$ consists in a simple modification of $\hat{\mu}_n^{(\text{NN-loo})}(f)$ by replacing $\hat{c}_{n,i}$, which involves $n-1$ Voronoi volumes, by $nV_{n,i}$. For Hölder integrands of order $s \in (0, 1]$, the difference between both estimates is of the order $n^{-1/2-s/d}$, in view of Eq. (3.3) below and Proposition 3.3 in the Appendix.

3.3 Main results

This section gathers some preliminary considerations on distributions on metric spaces and a bound on moments of nearest-neighbor distances before stating the main theoretical properties of the control neighbor estimates (3.1) and (3.2). Under suitable conditions on the measure μ and the integrand f , their root mean squared errors are shown to reach the rate $E[|\hat{\mu}_n(f) - \mu(f)|^2]^{1/2} \lesssim n^{-1/2}n^{-s/d}$, where two positive sequences a_n and b_n satisfy $a_n \lesssim b_n$ provided $a_n \leq Cb_n$ for some constant $C > 0$. Further, their finite-sample performance is studied through a concentration inequality providing a high-probability bound on the error $|\hat{\mu}_n(f) - \mu(f)|$.

3.3.1 Distributions on metric spaces

Let μ be a Borel probability measure on a bounded metric space (M, ρ) with diameter $\text{diam}(M)$. To control the nearest neighbor distances, we need a handle on the probabilities of small balls. Let $B(x, r) = \{y \in M : \rho(x, y) \leq r\}$ denote the ball with center $x \in M$ and radius $r > 0$. Intuitively, if μ is an absolutely continuous measure on a compact subset of $\mathbb{R}^d$, then $\mu(B(x, r))$ is of the order r^d . The same is true if μ is a smooth measure on a d -dimensional manifold. These considerations motivate the following assumption, which implicitly defines a kind of intrinsic dimension of M as measured by μ .

(A1) There exist $d > 0$ and $0 < C_0 \leq C_1 < \infty$ such that

$$\forall x \in M, \forall r \in (0, \text{diam}(M)], \quad C_0 r^d \leq \mu(B(x, r)) \leq C_1 r^d.$$

Remark 3.1. In (A1), it is sufficient to require the stated inequalities for all $r \in (0, r_0]$ for some $r_0 > 0$. Upon changing C_0 and C_1 , the inequalities then hold for all $r \in (r_0, \text{diam}(M)]$ too. Indeed, for such r , we then have the upper bound $\mu(B(x, r)) \leq 1 \leq r_0^{-d} \cdot r^d$ and the lower bound $\mu(B(x, r)) \geq \mu(B(x, r_0)) \geq C_0 r_0^d \geq C_0 \{r_0 / \text{diam}(M)\}^d \cdot r^d$.

Remark 3.2. In theory, the exponent d in (A1) could be non-integer. As an example, suppose μ is the normalized Hausdorff measure on a fractal-like set, like the Cantor set ($d = \log 2 / \log 3$) or the Sierpinski triangle ($d = \log 3 / \log 2$), see Stein and Shakarchi (2005). For the integration problem we study, this does not seem of much practical interest.

A sufficient hypothesis for (A1) is that μ is supported by a bounded subset of $\mathbb{R}^d$ with positive d -dimensional Lebesgue measure and is absolutely continuous with a well-behaved density:

(A1') The measure μ is supported by a compact subset M of $\mathbb{R}^d$ and admits a density p with respect to the d -dimensional Lebesgue measure λ_d . There exist constants $b, U, c, r_0 \in (0, \infty)$ with $b \leq U$ such that

- $\forall x \in M, \quad b \leq p(x) \leq U;$
- $\forall r \in (0, r_0], \forall x \in M, \quad \lambda_d(M \cap B(x, r)) \geq c \lambda_d(B(x, r)).$

Assumption (A1') is related to the *strong density assumption* of Audibert and Tsybakov (2007). It ensures that the density p of the measure μ has a sufficiently regular support and that it is bounded away from zero and infinity. In view of Remark 3.1, it is easy to see that (A1') implies (A1): since $\mu(B(x, r)) = \int_{M \cap B(x, r)} p(y) dy$, we have, for $x \in M$ and $r \in (0, r_0]$, the bounds

$$\begin{aligned} \mu(B(x, r)) &\leq U \lambda_d(B(x, r)) \leq U V_d \cdot r^d, \quad \text{and} \\ \mu(B(x, r)) &\geq b \lambda_d(M \cap B(x, r)) \geq bc \lambda_d(B(x, r)) \geq bc V_d \cdot r^d, \end{aligned}$$

where $V_d = \lambda_d(B(0, 1))$ is the volume of the unit ball in $\mathbb{R}^d$.

Another case of interest is when μ is supported by a subset of a closed d -dimensional Riemannian manifold $(\mathcal{M}_g, g)$. Let ρ_g be the geodesic distance on $\mathcal{M}_g$ induced by the metric g and let $B_g(x, r) = \{y \in \mathcal{M}_g : \rho_g(x, y) \leq r\}$ denote the geodesic ball with center $x \in \mathcal{M}_g$ and radius $r > 0$.

(A1'') The measure μ admits a density p having support $M = \{x \in \mathcal{M}_g : p(x) > 0\}$ with respect to the Riemannian volume measure ν_g on $\mathcal{M}_g$. The sectional curvatures of $\mathcal{M}_g$ are not greater than some $\delta > 0$ and there exist constants $b, U, c, r_0 \in (0, \infty)$ with $b \leq U$ such that

- $\forall x \in M, \quad b \leq p(x) \leq U;$
- $\forall 0 < r \leq r_0, \forall x \in M, \quad v_g(\mathcal{M}_g \cap B_g(x, r)) \geq c v_g(B_g(x, r)).$

Since $\mu(B_g(x, r)) = \int_{M \cap B_g(x, r)} p(y) v_g(dy)$, the argument that (A1'') implies (A1) is the same as the one of the sufficiency of (A1') above, provided we can relate the volume of a geodesic ball to its radius. This is the content of the next result.

Theorem (Corollary 2.2 in Kokarev (2021)). *Let $(\mathcal{M}_g, g)$ be a closed d -dimensional Riemannian manifold whose sectional curvatures are not greater than δ , where $\delta > 0$. Let $\text{inj}(\mathcal{M}_g)$ denote the injectivity radius of $\mathcal{M}_g$. Then, for any point $x \in \mathcal{M}_g$ and any positive radius $r \leq \text{rad}(g)$, the volume of the geodesic ball $B_g(x, r)$ satisfies the inequalities*

$$2^{1-d} V_d r^d \leq v_g(B_g(x, r)) \leq 2^{d-1} \frac{v_g(\mathcal{M}_g)}{\text{rad}(g)^d} r^d,$$

where V_d is the volume of a unit ball in the d -dimensional Euclidean space, $\text{rad}(g)$ stands for $\min\{\text{inj}(\mathcal{M}_g), \pi/(2\sqrt{\delta})\}$.

In Section 3.4, we perform numerical experiments featuring manifolds such as the unit sphere $\mathbb{S}^{m-1}$ in $\mathbb{R}^m$ and the orthogonal group. The cited theorem is valid for $\mathbb{S}^{m-1}$ since the sectional curvature of a $(m-1)$ -sphere of radius r is $1/r^2$ (Lee, 2019). The orthogonal group is a special case of the Stiefel manifold, which has a positive bounded sectional curvature (Zimmermann and Stoye, 2024). Other examples are the complex projective space $\mathbb{CP}^d$ and the quaternion projective space $\mathbb{HP}^d$, both having sectional curvatures bounded between 1 and 4 (Ichida, 2000).

3.3.2 Moments of nearest-neighbor distance

Consider drawing independent random samples from a distribution on a metric space.

- (A2) $X, X_1, X_2, \dots$ are independent and identically distributed random variables in a bounded metric space (M, ρ) with common distribution μ .

For Hölder-continuous functions, the quality of the approximation of the integrand f by its nearest-neighbor estimate $\hat{f}_n$ depends on the nearest-neighbor distance $\hat{\tau}_n$ (see Section 3.2.1). The next lemma provides control on the k -nearest neighbor distance $\hat{\tau}_{n,k}(x)$ and is inspired by the k -NN literature (Biau and Devroye, 2015). Let Γ denote the Euler gamma function.

Lemma 3.3 (Bounding moments of nearest neighbor distances). *Under (A1) and (A2), we have, for any $q > 0$ and for integer $k \in \{1, \dots, n\}$,*

$$\mathbb{E} [\hat{\tau}_{n,k}(x)^q] \leq (C_0 n)^{-q/d} \frac{\Gamma(k + q/d)}{\Gamma(k)}.$$

Remark 3.3 (Lower bounds). *In Lemma 3.3, we only need the lower bound in (A1), not the upper bound. This uniform lower bound in Condition (A1) plays an important role in our analysis as it allows a uniform control on the Voronoi cell diameters. Some refinement might be possible given the recent progress in k -NN regression for covariates with unbounded support (Kohler et al., 2006) and k -NN classification using some tail assumption on the covariates (Gadat et al., 2016). Extending the present analysis to such general measures is left for further research.*

3.3.3 Root mean squared error bounds

The main result of this section provides a finite-sample bound on the root mean squared error of the two control neighbors estimates in (3.1) and (3.2). The bound depends on the regularity of the integrand.

(A3) The function $f : M \rightarrow \mathbb{R}$ is Hölder continuous, i.e., there exist $s \in (0, 1]$ and $L > 0$ such that

$$\forall x, y \in M, \quad |f(x) - f(y)| \leq L\rho(x, y)^s.$$

The leave-one-out version $\hat{\mu}_n^{(\text{NN-loo})}$ involves n additional integrals and is computationally cumbersome. The proposed estimate $\hat{\mu}_n^{(\text{NN})}$ requires only a single additional integral while remaining close to the leave-one-out estimate, since their difference is

$$\hat{\mu}_n^{(\text{NN})}(f) - \hat{\mu}_n^{(\text{NN-loo})}(f) = \mu(\hat{f}_n) - \frac{1}{n} \sum_{i=1}^n \mu(\hat{f}_n^{(i)}). \quad (3.3)$$

Using this property and Lemma 3.1, we obtain that the root mean squared distance between the leave-one-out version and the proposed estimate is of the order $\mathcal{O}(n^{-1/2-s/d})$ as $n \rightarrow \infty$; see Proposition 3.3 in the Appendix for a precise statement. Therefore, the two estimates share the same convergence rate. In the next theorem, we assume (A1), for which each of (A1') and (A1'') is a sufficient condition.

Theorem 3.1 (Root mean squared error bounds). *Under (A1), (A2) and (A3), if $n \geq 4$, then*

$$\begin{aligned} \mathbb{E} \left[\left| \hat{\mu}_n^{(\text{NN-loo})}(f) - \mu(f) \right|^2 \right]^{1/2} &\leq C_{\text{NN-loo}} n^{-1/2} n^{-s/d}, \\ \mathbb{E} \left[\left| \hat{\mu}_n^{(\text{NN})}(f) - \mu(f) \right|^2 \right]^{1/2} &\leq C_{\text{NN}} n^{-1/2} n^{-s/d}, \end{aligned}$$

where

$$\begin{aligned} C_{\text{NN-loo}} &= LC_0^{-s/d} (C_1/C_0)^{1/2} 2^{s/d} \sqrt{10\Gamma(2s/d + 2)}, \\ C_{\text{NN}} &= LC_0^{-s/d} \sqrt{\Gamma(2s/d + 2)} \left(2\sqrt{s/d + 1} + 2^{s/d} (C_1/C_0)^{1/2} \sqrt{10} \right). \end{aligned}$$

For Lipschitz functions ($s = 1$), the rates obtained in Theorem 3.1 match the complexity rate stated in Novak (2016), see Section 3.1. The results in the aforementioned paper are concerned about a slightly more precise context as they assert that no random integration rule (see the article for more details) can reach a higher accuracy—measured in terms of mean-square error—than $O(n^{-1-2/d})$ when the integration measure is the uniform distribution on the unit cube and the integrand is Lipschitz. Theorem 3.1 states that the optimal rate is in fact achieved for more general spaces and measures.

The control neighbors estimates involve an approximation of f by piecewise constant functions $\hat{f}_n^{(i)}$ and therefore cannot achieve an acceleration of the convergence rate of more than $n^{-1/d}$. It is an interesting question whether a higher-order nearest neighbor approximation can achieve a greater accuracy for integrands that are smoother than just Lipschitz. The simulation experiments in Section 3.4 will give some feeling on the actual improvement of the control neighbors estimate over the basic Monte Carlo method at finite sample sizes.

Remark 3.4. *As mentioned in Section 3.2.2, the control neighbor estimate $\hat{\mu}_n^{(\text{NN})}$ is in general not unbiased, in contrast to the leave-one-out version $\hat{\mu}_n^{(\text{NN-loo})}$. Still, by the first statement of Proposition 3.3 in the Appendix, the bias of $\hat{\mu}_n^{(\text{NN})}$ is of the order $n^{-1-s/d}$ and thus constitutes an asymptotically negligible part of the estimator's root mean squared error, which decreases with rate $n^{-1/2-s/d}$.*

3.3.4 Concentration inequalities

In order to obtain a finite-sample performance guarantee of the proposed estimates, we apply an extension of McDiarmid's concentration inequality for functions with bounded differences on a high probability set A . The inequality is stated in Theorem 3.3 in Section 3.6.1 in the Appendix and is itself a minor extension of an inequality due to Combes (2015).

Because of our reliance on theory for nearest neighbor processes from Portier (2023), our bound is established only for measures supported by a compact subset M of Euclidean space, which is why we assume the more specific setting of (A1') rather than (A1).

Assumption (A3) implies that f is uniformly bounded on M , since M has a finite diameter. Write $C_f = \frac{1}{2} \{ \sup_{x \in M} f(x) - \inf_{x \in M} f(x) \}$. The two control neighbor estimates satisfy the following concentration inequalities.

Theorem 3.2 (Concentration inequalities). *Under (A1'), (A2) and (A3), we have, for any $\varepsilon \in (0, 1)$, with probability at least $1 - \varepsilon$,*

$$\begin{aligned} \left| \hat{\mu}_n^{(\text{NN-loo})}(f) - \mu(f) \right| &\leq K_1 \Delta_{n,\varepsilon} + \frac{6C_f}{n^{1/2+s/d}} \quad \text{and} \\ \left| \hat{\mu}_n^{(\text{NN})}(f) - \mu(f) \right| &\leq K_2 \Delta_{n,\varepsilon} + \frac{6C_f}{n^{1/2+s/d}} + \frac{(s/d + 2)L(V_d b c)^{-s/d}}{n^{1+s/d}}, \end{aligned}$$

where

$$\Delta_{n,\varepsilon} = \frac{\sqrt{\log(2/\sqrt{\varepsilon})} + \frac{1}{n^{s/d}}}{n^{1/2+s/d}} \cdot \begin{cases} \log(24n/\varepsilon)^{1+s/d} & \text{for } \varepsilon \leq \frac{2}{n^{1/2+s/d}}, \\ (3 \log(3n))^{1+s/d} & \text{for } \varepsilon > \frac{2}{n^{1/2+s/d}}. \end{cases}$$

The values of $K_1, K_2 > 0$ depend on (b, c, d, L, s, U) and are provided explicitly in the proof.

For small but fixed $\varepsilon \in (0, 1)$, the concentration inequalities state that the probability that the error $|\hat{\mu}_n(f) - \mu(f)|$ is larger than a constant multiple of $\sqrt{\log(1/\varepsilon)}(\log n)^{1+s/d}n^{-1/2}n^{-s/d}$ is less than ε . The additional logarithmic factor in Theorem 3.2 in comparison to Theorem 3.1 comes from the need to exert uniform control on the largest nearest neighbor distance via Lemma 3.5 in Section 3.6.1 in the Appendix, which in turn originates from a concentration inequality in Portier (2023).

3.3.5 Optimal L_p -approximation with nearest neighbors

The upper bounds on nearest neighbor distances stated in Lemma 3.3 have been key to examine the control neighbor estimators $\hat{\mu}_n^{(\text{NN-loo})}(f)$ and $\hat{\mu}_n^{(\text{NN})}(f)$ as claimed in Theorems 3.1 and 3.2. Another application of these upper bounds is the L_p -approximation of the nearest neighbor function estimator $\hat{f}_{n,k}(x) = k^{-1} \sum_{i=1}^k f(\hat{N}_{n,i}(x))$ of $f(x)$. Note that this type of estimator is often used in the setting of noiseless regression where one has access to $f(X_i)$ without noise. A precise statement is given in the next proposition, the proof of which is given in the Appendix.

Proposition 3.2 (L_p -approximation with nearest neighbors). *Under (A1), (A2), and (A3), we have, for any $k \in \{1, \dots, n\}$ and $p \geq 1$,*

$$\forall x \in M, \quad \mathbb{E} \left[\left| \hat{f}_{n,k}(x) - f(x) \right|^p \right]^{1/p} \leq L(C_0 n)^{-s/d} \left(\frac{\Gamma(k + sp/d)}{\Gamma(k)} \right)^{1/p}.$$

Since $\Gamma(x + \alpha) \sim x^\alpha \Gamma(x)$ as $x \rightarrow \infty$ for fixed $\alpha \in \mathbb{R}$, the upper bound is of the order $(k/n)^{s/d}$ as $k = k_n \rightarrow \infty$. We emphasize that similar results as in Proposition 3.2 can be found among the noiseless regression literature. However, to the best of our knowledge, most of the results seem to have been obtained for Euclidean covariates, i.e., when $M = \mathbb{R}^d$. In Chapter 15 in Biau and Devroye (2015), the weak convergence of $n^{1/d}(\hat{f}_{n,k}(x) - f(x))$ is established. In Theorem 1 from Kohler and Krzyżak (2013), the same rate of convergence for the integrated error is obtained. Some other results are given in Hinrichs et al. (2020); Krieg et al. (2022) where the authors investigate whether the distribution of the points can deteriorate the L_p -error or not. They show that the optimal rate of convergence can be obtained using either uniformly distributed points or deterministic points from the uniform grid. Our Proposition 3.2 confirms, not only for the Euclidean space but for general metric spaces, the following principle, that when there are enough points everywhere, as implied by (A1), the optimal rates of convergence for the L_p -error can be achieved.

Even though the analysis of the control neighbor estimate cannot be deduced from Proposition 3.2 because of the probabilistic dependence that is introduced in the algorithm (see the proofs of Theorems 3.1 and 3.2), the previous statement can be used for a better understanding of the control neighbor estimator as it shows that the rate obtained in Theorem 3.1 can be decomposed as the usual Monte Carlo error, $n^{-1/2}$, times the L_p -approximation error of the control variates estimating the integrand, $n^{-s/d}$. This is in line with the results from Portier and Segers (2018) where the same type of decomposition can be found for another control variates based estimator.

3.4 Numerical experiments

In Section 3.4.1, we start by presenting the *control neighbors* algorithm along with some practical remarks related to its implementation. Next, we present in Section 3.4.2 some examples involving integration problems on different spaces. Finally, Section 3.4.3 deals with the application of Monte Carlo estimates for optimal transport when computing the Sliced-Wasserstein distance. Additional experiments on a finance application with Monte Carlo exotic option pricing are available in Section 3.6.5 in the Appendix. To ensure repro-

ducibility of the numerical results, the Python code is available online¹.

3.4.1 Practical implementation

The procedure for computing the control neighbor estimate $\hat{\mu}_n^{(\text{NN})}(f)$ is presented in Algorithm 3.1. Required is a collection of points $X_1, \dots, X_n$ generated from a distribution μ . The estimate is based on the evaluations $f(X_i)$ of the integrand and the evaluations $\hat{f}_n^{(i)}(X_i)$ of the leave-one-out 1-nearest neighbors (1-NN) estimates of $f(X_i)$. Here, $\hat{f}_n^{(i)}$ denotes the 1-NN estimate of f constructed from the sample $\mathcal{X}_n^{(i)} = \{X_1, \dots, X_n\} \setminus \{X_i\}$ without the i -th particle, for $i \in \{1, \dots, n\}$; see Section 3.2.1 for precise definitions. Also required is the integral $\mu(\hat{f}_n)$ of the 1-NN estimate $\hat{f}_n$ based on the whole sample. Several practical remarks regarding the computation of all these quantities are given next.

Algorithm 3.1 Control Neighbors for Monte Carlo integration

Require: integrand f , probability measure μ , number of particles n .

- 1: Generate an independent random sample $X_1, \dots, X_n$ from μ
 - 2: Compute evaluations $f(X_1), \dots, f(X_n)$
 - 3: Compute nearest neighbor evaluations $\hat{f}_n^{(1)}(X_1), \dots, \hat{f}_n^{(n)}(X_n)$
 - 4: Compute the integral of the nearest neighbor estimate $\mu(\hat{f}_n)$
 - 5: Return $\frac{1}{n} \sum_{i=1}^n [f(X_i) - \{\hat{f}_n^{(i)}(X_i) - \mu(\hat{f}_n)\}]$
-

Nearest neighbors and tree search. The naive neighbor search implementation involves the computation of distances between all pairs of points in the training samples and may be computationally prohibitive. To address such practical inefficiencies, a variety of tree-based data structures have been invented to reduce the cost of a nearest neighbors search. The KD-Tree (Bentley, 1975) is a binary tree structure which recursively partitions the space along the coordinate axes, dividing it into nested rectangular regions into which data points are filed. The construction of such a tree requires $O(dn \log n)$ operations (Friedman et al., 1977). Once constructed, the query of a nearest neighbor in a KD-Tree can be done in $O(\log n)$ operations. However, in high dimension, the query cost increases and the structure of Ball-Tree (Omohundro, 1989) is favoured. Where KD trees partition data points along the Cartesian axes, Ball trees do so in a series of nested hyper-spheres, making tree construction more costly than for a KD tree, but resulting in an efficient data structure even in high dimensions. Many software libraries contain KD-tree

¹<https://github.com/RemiLELUC/ControlNeighbors>

and Ball-Tree implementations with efficient compression and parallelization (Pedregosa et al., 2011; Johnson et al., 2019).

Evaluation of $\hat{f}_n^{(i)}(X_i)$. When the computing time is measured through the evaluation of the integrand, the n additional evaluations $\hat{f}_n^{(i)}(X_i)$ are not computationally difficult as no additional calls to f are necessary. The leave-one-out nearest neighbor evaluation can be easily obtained as follows: first fit a KD-Tree on the particles $X_1, \dots, X_n$, then query the 2-nearest neighbor of each X_i to produce the vector of leave-one-out nearest neighbors.

Evaluation of $\mu(\hat{f}_n)$ and Voronoi volumes. The quantity $\mu(\hat{f}_n)$ is the sum of the evaluations $f(X_i)$ weighted by the values of the Voronoi volumes associated to the sample points X_i (see Definition 3.2 in the preceding section). The Voronoi volumes may be hard to compute but can always be approximated. The integral $\mu(\hat{f}_n)$ of the nearest neighbor estimate may be replaced by a Monte Carlo approximation based on N particles, such as $\tilde{\mu}_N(\hat{f}_n) = N^{-1} \sum_{i=1}^N \hat{f}_n(\tilde{X}_i)$, where the variables $\tilde{X}_i$ are drawn independently from μ . No additional evaluations of f are required. Conditionally on the first sample $X_1, \dots, X_n$, the error of this additional Monte Carlo approximation is $O(N^{-1/2})$, meaning that large values of the form $N = n^{1+2/d}$ permit to preserve the $O(n^{-1/2}n^{-1/d})$ rate of the control neighbors estimate for Lipschitz functions.

In case the measure μ is the uniform measure on $[0, 1]^d$, one may be able to explicitly compute the Voronoi volumes. Starting from the pioneering work of Richards (1974) in the context of protein structures, there have been advances to perform efficient Voronoi volume computations using Delaunay triangulations and taking advantage of graphic hardware (Hoff III et al., 1999). Computations for Voronoi tessellations in dimensions $d = 2$ and $d = 3$ are implemented in the Voronoi++ software (Rycroft, 2009). However, this type of algorithm might become inefficient when d is large.

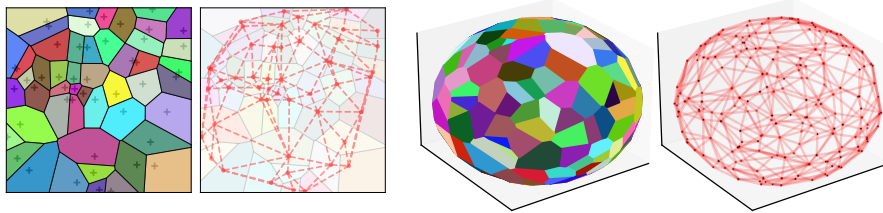


Figure 3.1: Voronoi cells and Delaunay triangulations on $[0, 1]^2$ and $\mathbb{S}^2$. (Made with Python package Matplotlib (Hunter, 2007))

Computing time. When using the standard KD-tree approach, the computation time of our algorithm can be estimated in the light of the previous remarks. The different computation times are: $O(nd \log n)$ for building the

KD-tree, $O(n \log n)$ for the evaluations $\{\hat{f}_n^{(i)}(X_i)\}_{i=1}^n$ and finally, $O(N \log n)$ operations for the estimation of $\mu(\hat{f}_n)$. Choosing N as recommended before, the overall complexity is $O((nd+n^{1+2/d}) \log n)$ operations. Note that when the Voronoi volumes are available, the computation time reduces to $O(nd \log n)$.

Extensions. A natural variant of the proposed method is obtained by replacing the 1-NN estimate $\hat{f}_n$ in Eq. (3.1) by a k -NN estimate $\hat{f}_n^{(k)}$ which averages the evaluations of the k nearest neighbors of a given point. The estimate is then defined by $\hat{f}_n^{(k)}(x) = k^{-1} \sum_{j=1}^k f(\hat{N}_{n,j}(x))$ where $\hat{N}_{n,j}(x)$ is the j -nearest neighbor of x . This involves both the tuning of the hyperparameter $k \geq 1$ and some extra computation due to the associated nearest neighbors search. In regression or classification, high values of k can reduce the variance of the estimate by averaging the model noise at the cost of added computations. In contrast, the control neighbors estimate ($k = 1$) is free of these additional costs and takes advantage of the noiseless evaluations (Biau and Devroye, 2015, Chapter 15) of the integrand.

3.4.2 Integration on various spaces: $[0, 1]^d, \mathbb{R}^d, O_m(\mathbb{R})$ and $\mathbb{S}^{q-1}$

The aim of this section is to empirically validate the $O(n^{-1/2}n^{-1/d})$ convergence rate of the control neighbors estimate in a wide variety of integration problems ranging from the unit cube $[0, 1]^d$, the group of orthogonal matrices $O_m(\mathbb{R})$, which is an $m(m-1)/2$ dimensional manifold embedded in $\mathbb{R}^{m \times m}$, and the unit sphere $\mathbb{S}^{q-1}$ in $\mathbb{R}^q$. In the different settings, the sample size evolves from $n = 10^1$ to $n = 10^4$. The figures report the evolution of the root mean squared error $n \mapsto E[|\hat{\mu}_n^{(\text{NN})}(f) - \mu(f)|^2]^{1/2}$, where the expectation is computed over 100 independent replications. In all experiments, MC represents the naive Monte Carlo estimate and CVNN returns the value of $\hat{\mu}_n^{(\text{NN})}(f)$ for which the integral $\mu(f_n)$ is replaced by a Monte Carlo estimate that uses n^2 particles, with n the number of evaluations of f .

Integration on $[0, 1]^d$ and $\mathbb{R}^d$. Consider first the integration problem $\mu(f) = \int f d\mu$ where the measure μ is either the uniform distribution over the unit cube $[0, 1]^d$ or the multivariate Gaussian measure on $\mathbb{R}^d$. The latter case is not covered by our theory, since the support is unbounded. The goal is to compute $\int f_1(x) \mathbb{1}_{[0,1]^d}(x) dx$ and $\int f_2(x) \psi(x) dx$ with

$$\begin{aligned} f_1(x_1, \dots, x_d) &= \sin \left(\pi \left(\frac{2}{d} \sum_{i=1}^d x_i - 1 \right) \right), \\ f_2(x_1, \dots, x_d) &= \sin \left(\frac{\pi}{d} \sum_{i=1}^d x_i \right) \end{aligned} \tag{3.4}$$

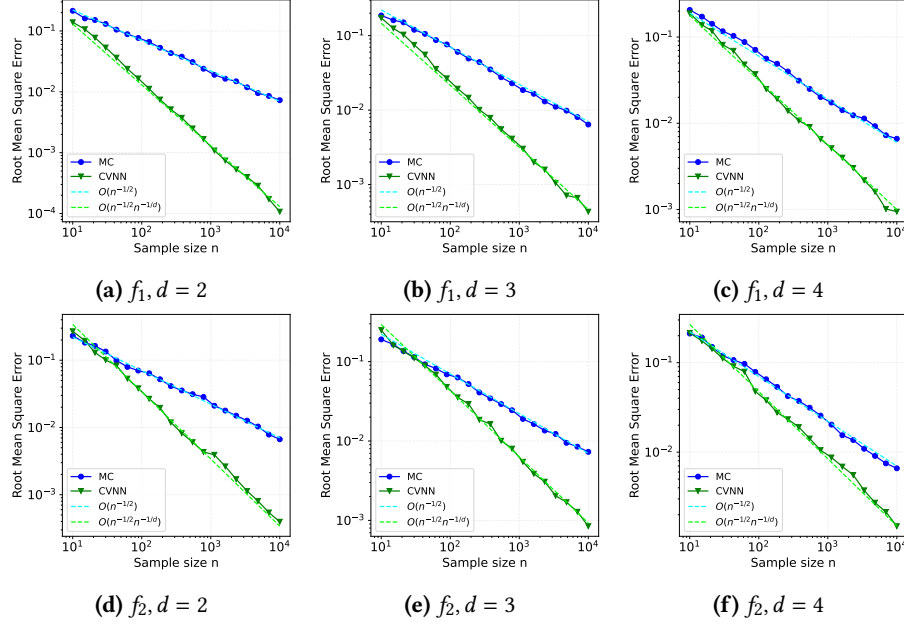


Figure 3.2: Root mean squared errors obtained over 100 replications for functions f_1 (top) and f_2 (bottom) in Eq. (3.4) in dimension $d \in \{2, 3, 4\}$ (left to right).

and with $\psi(\cdot)$ the probability density function of the multivariate Gaussian distribution $\mathcal{N}(0, I_d)$. Figure 3.2 displays the evolution of the root mean squared error for the two integrals in dimensions $d \in \{2, 3, 4\}$. The different error curves confirm the optimal convergence rate $O(n^{-1/2}n^{-1/d})$ for the control neighbors estimate and illustrate the error reduction with respect to the Monte Carlo method.

Integration on the orthogonal group $O_m(\mathbb{R})$. Consider the group of real orthogonal matrices $O_m(\mathbb{R}) = \{X \in \text{GL}_m(\mathbb{R}) : X^\top X = XX^\top = I_m\}$. As a toy example, we compute moments of the trace of a random orthogonal matrix, i.e., integrands of the form $O_m(\mathbb{R}) \rightarrow \mathbb{R} : X \mapsto \text{tr}(X)^k$, when X is drawn uniformly on $O_m(\mathbb{R})$. In fact, exact expressions are known; see for instance Diaconis and Evans (2001) and Pastur and Vasilchuk (2004) for background and results. The goal is thus to compute the integrals

$$a_k = \int_{O_m(\mathbb{R})} \text{tr}(X)^k d\mu(X), \quad (3.5)$$

with $d\mu(X)$ the unit Haar measure on $O_m(\mathbb{R})$. Assumption (A1) is satisfied with $d = m(m-1)/2$. In practice, random orthogonal matrices $X_1, \dots, X_n \in O_m(\mathbb{R})$ are generated using the function `ortho_group` of the Python package `scipy` (Virtanen et al., 2020). This function returns random orthogonal matri-

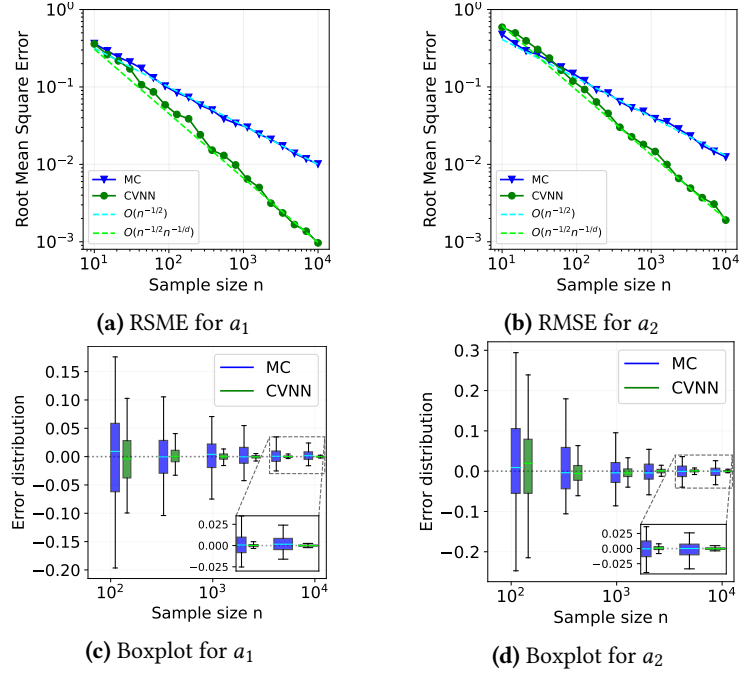


Figure 3.3: Root mean squared errors (left) and boxplots of the errors (right) obtained over 100 replications for integrands $X \mapsto \text{tr}(X)^k$ for $k \in \{1, 2\}$ in Eq. (3.5) with respect to the unit Haar measure on the orthogonal group $O_3(\mathbb{R})$.

ces drawn from the Haar distribution using a careful QR decomposition² as in Mezzadri (2007). The nearest neighbors are computed using the norm associated to the Frobenius inner product $\langle X, Y \rangle = \text{tr}(X^\top Y)$ for $X, Y \in O_m(\mathbb{R})$. Figure 3.3 reports the evolution of the root mean squared error and boxplots of the integration error over the group $O_3(\mathbb{R})$ which corresponds to dimension $d = 3$. Both integrands are continuously differentiable and thus Lipschitz, since $O_m(\mathbb{R})$ is compact. Once again, the experiments empirically validate the convergence rates of the Monte Carlo methods and reveal the variance reduction obtained with the control neighbors estimate.

Integration on the sphere $\mathbb{S}^2$. The unit sphere $\mathbb{S}^2$ in $\mathbb{R}^3$ is a compact Riemannian manifold of dimension $d = 2$ with positive curvature. The distances on the sphere are computed using the equations of the great circles. Consider

²QR decomposition refers to the factorization of a matrix X into a product $X = QR$ of an orthonormal matrix Q and an upper triangular matrix R .

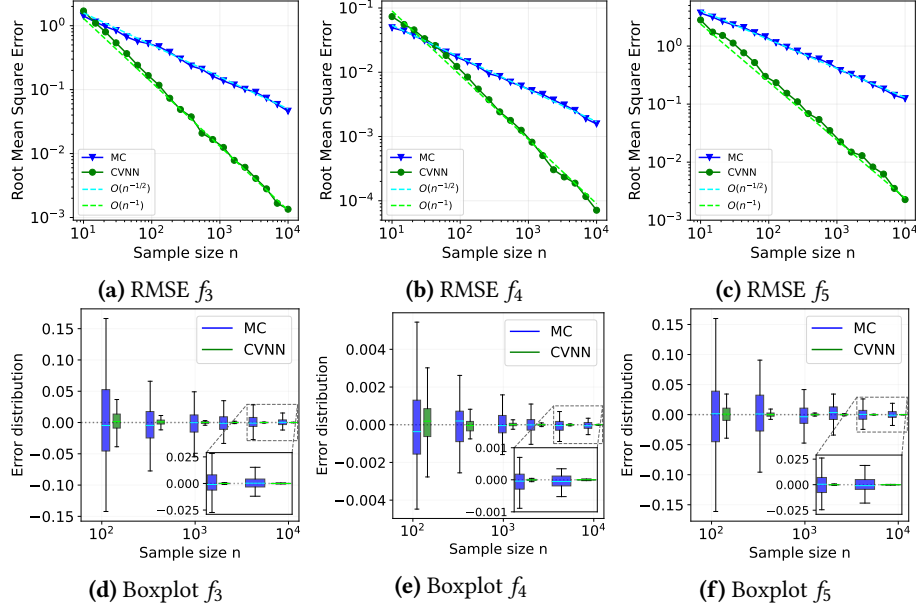


Figure 3.4: Root mean squared errors (top) and boxplots of the errors (bottom) obtained over 100 replications for functions f_3 (left), f_4 (middle) and f_5 (right) in Eq. (3.6) when integrating over $\mathbb{S}^2$.

the integral $\int_{\mathbb{S}^2} f \, d\Omega$ with Ω the uniform distribution on $\mathbb{S}^2$ and integrands

$$\begin{aligned} f_3(x, y, z) &= \cos(x + y + z), \\ f_4(x, y, z) &= \cos(x) \cos(y) \cos(z), \\ f_5(x, y, z) &= \exp(x - y), \end{aligned} \tag{3.6}$$

so that $\mu(f_3) = \mu(f_4) = (4\pi/\sqrt{3}) \sin(\sqrt{3})$ and $\mu(f_5) = \pi\sqrt{8} \sinh(\sqrt{2})$. Figure 3.4 reports the evolution of the root mean squared error and the boxplots of the errors over the sphere $\mathbb{S}^2$. The error curves empirically validate the $O(n^{-1})$ convergence rate for the control neighbors estimate while the boxplots highlight the variance reduction of the proposed method.

3.4.3 Monte Carlo integration for optimal transport

Optimal transport. Optimal transport (OT) is a mathematical framework for measuring the distance between probability distributions. It has proven to be particularly useful in machine learning applications, where it can be used for tasks such domain adaptation (Courty et al., 2017) and image generation (Gulrajani et al., 2017; Genevay et al., 2018). The recent development of efficient algorithms for solving optimal transport problems, e.g., the Sliced-Wasserstein (SW) distance (Rabin et al., 2012) and Sinkhorn distance (Cuturi,

2013), has made it a practical tool for large-scale machine learning applications.

OT distances. For $X \subseteq \mathbb{R}^q$, let $\mathcal{P}(X)$ denote the set of probability measures supported on X and let $p \in [1, \infty)$. The Wasserstein distance of order p between $P, Q \in \mathcal{P}(X)$ is

$$W_p^p(P, Q) = \inf_{\pi \in \Pi(P, Q)} \int_{X \times X} \|x - y\|^p d\pi(x, y),$$

where $\Pi(P, Q) \subset \mathcal{P}(X \times X)$ denotes the set of couplings for (P, Q) , i.e., probability measures whose marginals with respect to the first and second variables are P and Q respectively. While the Wasserstein distance enjoys attractive theoretical properties (Villani, 2009, Chapter 6), it suffers from a high computational cost. When computing $W_p(P_m, Q_m)$ for discrete distributions P_m and Q_m supported on m points, the worst-case computational complexity scales as $O(m^3 \log m)$ (Peyré et al., 2019). To overcome this issue, the Sliced-Wasserstein distance takes advantage of the fast computation of the Wasserstein distance between univariate distributions $P, Q \in \mathcal{P}(\mathbb{R})$. Indeed, for $P_m = (1/m) \sum_{i=1}^m \delta_{x_i}$ and $Q_m = (1/m) \sum_{i=1}^m \delta_{y_i}$ with $x_i, y_i \in \mathbb{R}$, the W_p -distance involves sorting the atoms $x_{(1)} \leq \dots \leq x_{(m)}$ and $y_{(1)} \leq \dots \leq y_{(m)}$, yielding

$$W_p^p(P_m, Q_m) = \frac{1}{m} \sum_{i=1}^m |x_{(i)} - y_{(i)}|^p,$$

leading to a complexity of $O(m \log m)$ operations induced by the sorting step.

Recall that $\mathbb{S}^{q-1} = \{\theta \in \mathbb{R}^q : \|\theta\| = 1\}$ is the unit sphere in $\mathbb{R}^q$ and for $\theta \in \mathbb{S}^{q-1}$ let $\theta^\star : \mathbb{R}^q \rightarrow \mathbb{R}$ denote the linear map $\theta^\star(x) = \langle \theta, x \rangle$ for $x \in \mathbb{R}^q$.

Let μ denote the uniform distribution on $\mathbb{S}^{q-1}$, that is, the normalized volume measure. The Sliced-Wasserstein (SW) distance (Rabin et al., 2012; Bonneel et al., 2015; Kolouri et al., 2019) of order p based on μ is defined for $P, Q \in \mathcal{P}(X)$ as

$$SW_p^p(P, Q, \mu) = \mathbb{E}_\theta[W_p^p(\theta^\star P, \theta^\star Q)] = \int_{\mathbb{S}^{q-1}} W_p^p(\theta^\star P, \theta^\star Q) d\mu(\theta), \quad (3.7)$$

where $g_\# \xi$ is the *push-forward* measure of $\xi \in \mathcal{P}(\mathbb{R}^q)$ by a measurable function g on $\mathbb{R}^q$. For $\theta \in \mathbb{S}^{q-1}$, the measures $\theta^\star P$ and $\theta^\star Q$ are the distributions of the projections $\langle \theta, X \rangle$ and $\langle \theta, Y \rangle$ of random vectors X and Y with distributions P and Q , respectively. The integrand is Lipschitz in view of Theorem 2.4 in Han (2023).

In practice, the random directions $\theta_1, \dots, \theta_n$ are sampled independently from the uniform distribution on the unit sphere $\mathbb{S}^{q-1}$ and the SW-distance

of Eq. (3.7) is approximated using a standard Monte Carlo estimate with n random projections as

$$\widehat{SW}_p^p(P_m, Q_m) = \frac{1}{n} \sum_{i=1}^n W_p^p((\theta_i)_\# P_m, (\theta_i)_\# Q_m).$$

The use of more accurate numerical integration methods to compute the SW-distance is an active area of current research; see, e.g., Nguyen and Ho (2024) and Leluc et al. (2024a). Below, we illustrate the benefits of using control neighbors, while at the same time we anticipate that more specialized methods may do even better for this particular integration problem.

Multivariate Gaussians. The goal is to compare the variance of the standard Monte Carlo estimate (SW-MC) with the proposed control neighbors estimate (SW-CVNN) when computing the Sliced-Wasserstein distance between two Gaussian distributions. More precisely, we want to compute $SW_2(P, Q)$, where $P = \mathcal{N}_q(m_X, \sigma_X^2 I_q)$ and $Q = \mathcal{N}_q(m_Y, \sigma_Y^2 I_q)$ with $m_X, m_Y \sim \mathcal{N}(0, I_q)$ and $\sigma_X = 2$ and $\sigma_Y = 5$. We consider the corresponding empirical distributions P_m and Q_m based on $m = 2000$ samples and compute the Monte Carlo estimates of the SW_2 distance using a number of projections $n \in \{50; 100; 250; 500; 1000\}$ in dimension $q \in \{3; 6\}$. In the notation of the general theory, we have $d = q - 1 \in \{2; 5\}$ since we are integrating over the $(q - 1)$ -dimensional sphere in $\mathbb{R}^q$. Figure 3.5 shows the error distribution of the different Monte Carlo estimates (SW-MC and SW-CVNN) of the SW_2 distance where the exact value (see Nadjahi et al., 2021, Appendix S3.1) is given by

$$SW_2^2(\mathcal{N}_q(m_X, \sigma_X^2 I_q), \mathcal{N}_q(m_Y, \sigma_Y^2 I_q)) = q^{-1} \|m_X - m_Y\|_2^2 + (\sigma_X - \sigma_Y)^2.$$

The different boxplots highlight the good performance of the control neighbors estimate in terms of variance reduction.

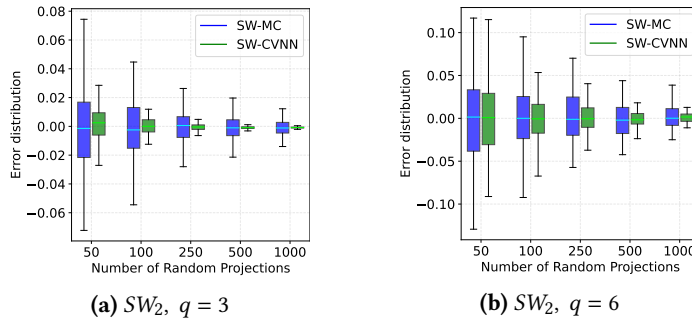


Figure 3.5: Boxplots of Sliced-Wasserstein estimates SW-MC and SW-CVNN for Gaussian distributions on $\mathbb{R}^q$ with $q \in \{3; 6\}$. The boxplots are obtained over 100 replications.

3.5 Discussion

We have explored the use of nearest neighbors in the construction of control variates for variance reduction in Monte Carlo integration. We have shown that for Hölder integrands of regularity $s \in (0, 1]$ on bounded metric spaces of dimension d as measured by a sufficiently regular probability distribution, a faster rate of convergence, namely $O(n^{-1/2}n^{-s/d})$ as $n \rightarrow \infty$, is possible through the construction of a control variate via leave-one-out neighbors. Theoretical guarantees are given both in terms of bounds on the root mean squared error and as concentration inequalities (requiring an additional logarithmic factor). In numerical experiments, the method enjoyed a notable error reduction with respect to Monte Carlo integration.

A drawback is that our method is not able to leverage higher-order smoothness of the integrand besides the Lipschitz property: this comes from the fact that the control variate is piecewise constant on the Voronoi cells induced by the Monte Carlo sample. As a consequence, the accuracy gain is limited on high-dimensional domains. Further, integration of the control function requires an additional Monte Carlo sample, of larger size than the original one, although without needing additional evaluations of the integrand.

3.6 Appendix

Section 3.6.1 contains some auxiliary results. Section 3.6.2 gathers the proofs of all the lemmas, while Section 3.6.3 is concerned with the proofs of the different propositions. Section 3.6.4 comprises the proofs of the theorems and Section 3.6.5 presents additional numerical experiments for option pricing.

3.6.1 Auxiliary results

The following theorem is a corollary to Theorem 2.1 in Combes (2015).

Theorem 3.3 (Extension of McDiarmid's inequality). *Let $X_1, \dots, X_n$ independent random variables with X_ℓ taking values in some measurable space Ω_ℓ . Define $X = (X_1, \dots, X_n) \in \prod_{\ell=1}^n \Omega_\ell =: \Omega$. Let $\phi : \Omega \rightarrow \mathbb{R}$ be a measurable function and assume there exists an event $A \subseteq \Omega$ and constants $c_1, \dots, c_n \geq 0$ such that, for any $\ell \in \{1, \dots, n\}$ and for any $x \in A$ and $x' \in A$ that differ only in the ℓ -th coordinate, we have*

$$|\phi(x) - \phi(x')| \leq c_\ell. \quad (3.8)$$

Then, for any $t \geq 0$, we have, writing $m = E[\phi \mid A]$, $p = 1 - P(A)$ and $\bar{c} = \sum_{\ell=1}^n c_\ell$,

$$P(|\phi(X) - m| \geq t) \leq p + 2 \exp\left(-\frac{2 \max(0, t - p\bar{c})^2}{\sum_{\ell=1}^n c_\ell^2}\right).$$

Lemma 3.4 (Kabatianskii and Levenshtein (1974); Zeger and Gersho (1994)). Given a point set $P \subset \mathbb{R}^d$ and $x \in P$, the maximum number of points in P that can have x as nearest neighbor is bounded by the d -dimensional kissing number ψ_d , where $\psi_d \leq 2^{0.401d(1+o(1))}$ as $d \rightarrow \infty$.

Proposition 3.3. Under Assumptions (A1), (A2) and (A3), if $n \geq 4$, we have

$$E \left[\left\| \frac{1}{n} \sum_{i=1}^n \mu(\hat{f}_n^{(i)}) - \mu(\hat{f}_n) \right\| \right] \leq (s/d + 2) L C_0^{-s/d} \Gamma(s/d + 1) n^{-1-s/d}$$

and

$$E \left[\left\| \frac{1}{n} \sum_{i=1}^n \mu(\hat{f}_n^{(i)}) - \mu(\hat{f}_n) \right\|^2 \right] \leq 4 (s/d + 1) L^2 C_0^{-2s/d} \Gamma(2s/d + 1) n^{-1-2s/d}.$$

Lemma 3.5 (Portier (2023)). Under Assumptions (A1') and (A2), for all $n \geq 1$, all $\delta \in (0, 1)$ and all $k \in \{1, \dots, n\}$ such that $8d \log(12n/\delta) \leq k \leq r_0^d nbcV_d/2$, it holds, with probability at least $1 - \delta$, that

$$\sup_{x \in \mathcal{X}} \hat{\tau}_{n,k}(x) \leq \left(\frac{2k}{nbcV_d} \right)^{1/d} =: \bar{\tau}_{n,k}.$$

Lemma 3.6 (Combes (2015)). Let $\phi : \Omega \rightarrow \mathbb{R}$ be a bounded, measurable function and let $A \subseteq \Omega$ be an event with $P(A) > 0$. Writing $m = E[\phi \mid A]$, $p = 1 - P(A)$ and $\|\phi\|_\infty = \sup_{x \in \Omega} |\phi(x)|$, we have

$$|E[\phi] - m| \leq 2p \|\phi\|_\infty.$$

Lemma 3.7. For $x, a > 0$, we have $\Gamma(x + 1 + a) > x^a \Gamma(x + 1)$.

3.6.2 Proofs of Lemmas

Proof of Lemma 3.1

Given any collection $X_1, \dots, X_n$ of n distinct points, if $j \neq i$, then $\hat{f}_n^{(i)}$ and $\hat{f}_n$ are the same on $S_{n,j}$. It holds that

$$\hat{f}_n^{(i)}(x) - \hat{f}_n(x) = \{\hat{f}_n^{(i)}(x) - \hat{f}_n(x)\} \mathbb{1}_{S_{n,i}}(x).$$

Now using that $\bar{f}_n$ and $\hat{f}_n^{(i)}$ are the same on $S_{n,i}$, it follows that

$$\hat{f}_n^{(i)}(x) - \hat{f}_n(x) = \{\bar{f}_n(x) - \hat{f}_n(x)\} \mathbb{1}_{S_{n,i}}(x).$$

Taking the sum and using $\sum_{i=1}^n \mathbb{1}_{S_{n,i}}(x) = 1$ gives

$$\sum_{i=1}^n \{\hat{f}_n^{(i)}(x) - \hat{f}_n(x)\} = \bar{f}_n(x) - \hat{f}_n(x),$$

and the result follows by integrating with respect to μ .

Proof of Lemma 3.2

Because the Voronoi cells define a partition of M , we have for any $x \in M$,

$$\hat{f}_n^{(i)}(x) = \sum_{j:j \neq i} f(X_j) \mathbb{1}_{S_{n,j}^{(i)}}(x)$$

and in particular

$$\hat{f}_n^{(i)}(X_i) = \sum_{j:j \neq i} f(X_j) \mathbb{1}_{S_{n,j}^{(i)}}(X_i),$$

from which we deduce

$$\sum_{i=1}^n \hat{f}_n^{(i)}(X_i) = \sum_{j=1}^n f(X_j) \sum_{i:i \neq j} \mathbb{1}_{S_{n,j}^{(i)}}(X_i) = \sum_{j=1}^n f(X_j) \hat{d}_{n,j}.$$

Further, we have

$$\mu(\hat{f}_n) = \sum_{i=1}^n \mu(f(X_i) \mathbb{1}_{S_{n,i}}) = \sum_{i=1}^n f(X_i) \mu(S_{n,i}) = \sum_{i=1}^n f(X_i) V_{n,i}$$

and

$$\sum_{i=1}^n \mu(\hat{f}_n^{(i)}) = \sum_{i=1}^n \sum_{j:j \neq i} \mu\left(f(X_j) \mathbb{1}_{S_{n,j}^{(i)}}\right) = \sum_{j=1}^n f(X_j) \sum_{i:i \neq j} V_{n,j}^{(i)} = \sum_{j=1}^n f(X_j) \hat{c}_{n,j}.$$

Proof of Lemma 3.3

For $x \in M$, $r \in (0, \text{diam}(M)]$ and integer $k \in \{1, \dots, n\}$, we have

$$\begin{aligned} \mathbb{P}[\hat{t}_{n,k}(x) > r] &= \mathbb{P}\left[\sum_{i=1}^n \mathbb{1}_{\{\rho(X_i, x) \leq r\}} \leq k-1\right] \\ &= \mathbb{P}[\text{Bin}(n, \mu(B(x, r))) \leq k-1] \\ &\leq \mathbb{P}[\text{Bin}(n, C_0 r^d) \leq k-1] \\ &= \sum_{j=0}^{k-1} \binom{n}{j} (C_0 r^d)^j (1 - C_0 r^d)^{n-j}. \end{aligned}$$

The equality remains valid for $r > 0$ such that $C_0 r^d \leq 1$, since if $r > \text{diam}(M)$, the probability is zero. We find

$$\begin{aligned}
 \mathbb{E} [\hat{\tau}_{n,k}(x)^q] &= \int_0^\infty \mathbb{P} [\hat{\tau}_{n,k}(x) > t^{1/q}] \, dt \\
 &\leq \sum_{j=0}^{k-1} \binom{n}{j} \int_0^{C_0^{-q/d}} (C_0 t^{d/q})^j (1 - C_0 t^{d/q})^{n-j} \, dt \\
 &= \sum_{j=0}^{k-1} \binom{n}{j} \int_0^1 u^j (1-u)^{n-j} \cdot \frac{q}{d} C_0^{-q/d} u^{q/d-1} \, du \\
 &= \frac{q}{d} C_0^{-q/d} \sum_{j=0}^{k-1} \binom{n}{j} \frac{\Gamma(j+q/d)\Gamma(n-j+1)}{\Gamma(n+q/d+1)},
 \end{aligned}$$

since the integral can be expressed in terms of the Beta function, which can in turn be developed in terms of the Gamma function. Writing the binomial coefficient in terms of the Gamma function too and simplifying yields

$$\begin{aligned}
 \mathbb{E} [\hat{\tau}_{n,k}(x)^q] &\leq \frac{q}{d} C_0^{-q/d} \frac{\Gamma(n+1)}{\Gamma(n+q/d+1)} \sum_{j=0}^{k-1} \frac{\Gamma(j+q/d)}{\Gamma(j+1)} \\
 &\leq \frac{q}{d} (C_0 n)^{-q/d} \sum_{j=0}^{k-1} \frac{\Gamma(j+q/d)}{\Gamma(j+1)},
 \end{aligned}$$

where we used Lemma 3.7 in the second inequality. By induction, one can obtain

$$\frac{q}{d} \sum_{j=0}^{k-1} \frac{\Gamma(j+q/d)}{\Gamma(j+1)} = \frac{\Gamma(k+q/d)}{\Gamma(k)}. \quad (3.9)$$

For $k = 1$, (3.9) reduces to

$$\frac{(q/d)\Gamma(q/d)}{\Gamma(1)} = \frac{\Gamma(q/d+1)}{\Gamma(1)}.$$

Assume that (3.9) holds for $k-1$, i.e.,

$$\frac{q}{d} \sum_{j=0}^{k-2} \frac{\Gamma(j+q/d)}{\Gamma(j+1)} = \frac{\Gamma(k-1+q/d)}{\Gamma(k-1)}. \quad (3.10)$$

Then, by (3.10) and $\Gamma(k) = (k-1)\Gamma(k-1)$, we have

$$\begin{aligned} \frac{q}{d} \sum_{j=0}^{k-1} \frac{\Gamma(j+q/d)}{\Gamma(j+1)} &= \frac{q}{d} \sum_{j=0}^{k-2} \frac{\Gamma(j+q/d)}{\Gamma(j+1)} + \frac{q}{d} \frac{\Gamma(k-1+q/d)}{\Gamma(k)} \\ &= \frac{\Gamma(k-1+q/d)}{\Gamma(k-1)} + \frac{q}{d} \frac{\Gamma(k-1+q/d)}{\Gamma(k)} \\ &= \frac{(k-1+q/d)\Gamma(k-1+q/d)}{\Gamma(k)} \\ &= \frac{\Gamma(k+q/d)}{\Gamma(k)}. \end{aligned}$$

Therefore,

$$\mathbb{E} [\hat{t}_{n,k}(x)^q] \leq (C_0 n)^{-q/d} \frac{\Gamma(k+q/d)}{\Gamma(k)}.$$

Proof of Lemma 3.7

The Gamma function is strictly logarithmically convex. Writing

$$x+1 = (1-\lambda)x + \lambda(x+1+a) \text{ with } \lambda = \frac{1}{1+a}$$

yields

$$\Gamma(x+1) < \Gamma(x)^{1-\lambda} \Gamma(x+1+a)^\lambda.$$

Since $\Gamma(x) = x^{-1}\Gamma(x+1)$, it follows that

$$\begin{aligned} \Gamma(x+1+a) &> \left\{ \Gamma(x+1)\Gamma(x)^{\lambda-1} \right\}^{1/\lambda} = \left\{ \Gamma(x+1)^{1+\lambda-1} x^{1-\lambda} \right\}^{1/\lambda} \\ &= \Gamma(x+1)x^{1/\lambda-1} = \Gamma(x+1)x^a, \end{aligned}$$

as required.

3.6.3 Proofs of Propositions

Proof of Proposition 3.1

Using Lemma 3.2, we find

$$\begin{aligned} \hat{\mu}_n^{(\text{NN})}(f) &= \frac{1}{n} \sum_{i=1}^n [f(X_i) - \{\hat{f}_n^{(i)}(X_i) - \mu(\hat{f}_n)\}] \\ &= \frac{1}{n} \sum_{i=1}^n f(X_i) - \frac{1}{n} \sum_{j=1}^n f(X_j) \hat{d}_{n,j} + \sum_{i=1}^n f(X_i) V_{n,i} \\ &= \frac{1}{n} \sum_{i=1}^n \left(1 - \hat{d}_{n,i} + nV_{n,i}\right) f(X_i) \end{aligned}$$

and, similarly,

$$\begin{aligned}
\hat{\mu}_n^{(\text{NN-loo})} &= \frac{1}{n} \sum_{i=1}^n [f(X_i) - \{\hat{f}_n^{(i)}(X_i) - \mu(\hat{f}_n^{(i)})\}] \\
&= \frac{1}{n} \sum_{i=1}^n f(X_i) - \frac{1}{n} \sum_{j=1}^n f(X_j) \hat{d}_{n,j} + \frac{1}{n} \sum_{j=1}^n f(X_j) \hat{c}_{n,j} \\
&= \frac{1}{n} \sum_{i=1}^n (1 - \hat{d}_{n,i} + \hat{c}_{n,i}) f(X_i),
\end{aligned}$$

as required.

Proof of Proposition 3.2

Let $N_{n,k}(x)$ denote the set of indices $i \in \{1, \dots, n\}$ such that X_i is among the k nearest neighbors of $x \in M$. Using the Hölder property and the definition of $\hat{\tau}_{n,k}$ we have, for each $i \in N_{n,k}(x)$,

$$|f(X_i) - f(x)| \leq L\rho(X_i, x)^s \leq L\hat{\tau}_{n,k}(x)^s.$$

It follows that

$$\left| \hat{f}_{n,k}(x) - f(x) \right| \leq \frac{1}{k} \sum_{i \in N_{n,k}(x)} |f(X_i) - f(x)| \leq L\hat{\tau}_{n,k}(x)^s.$$

Next we apply Lemma 3.3 with $q = sp$ to obtain

$$\begin{aligned}
\mathbb{E}[|\hat{f}_{n,k}(x) - f(x)|^p] &\leq L^p \mathbb{E}[\hat{\tau}_{n,k}(x)^{sp}] \\
&\leq L^p (C_0 n)^{-sp/d} \frac{\Gamma(k + sp/d)}{\Gamma(k)}.
\end{aligned}$$

Raising both sides to the power $1/p$ yields the stated inequality.

Proof of Proposition 3.3

Using the fact that $\hat{f}_n^{(i)}$ and $\hat{f}_n$ coincide outside $S_{n,i}$ and that $\hat{f}_n(x) = f(X_i)$ for $x \in S_{n,i}$, we have

$$\begin{aligned}
\mu(\hat{f}_n^{(i)}) - \mu(\hat{f}_n) &= \mu(\hat{f}_n^{(i)} - \hat{f}_n) \\
&= \mu((\hat{f}_n^{(i)} - \hat{f}_n) \mathbb{1}_{S_{n,i}}) \\
&= \mu(\{\hat{f}_n^{(i)} - f(X_i)\} \mathbb{1}_{S_{n,i}}).
\end{aligned}$$

Write $R_n = \left| \frac{1}{n} \sum_{i=1}^n \{ \mu(\hat{f}_n^{(i)}) - \mu(\hat{f}_n) \} \right|$. Then

$$\begin{aligned}
 R_n &\leq \frac{1}{n} \sum_{i=1}^n \left| \mu(\hat{f}_n^{(i)}) - \mu(\hat{f}_n) \right| \\
 &= \frac{1}{n} \sum_{i=1}^n \left| \mu(\{ \hat{f}_n^{(i)} - f(X_i) \} \mathbb{1}_{S_{n,i}}) \right| \\
 &\leq \frac{1}{n} \sum_{i=1}^n \int_{S_{n,i}} \left| \hat{f}_n^{(i)}(x) - f(X_i) \right| d\mu(x) \\
 &= \frac{1}{n} \sum_{i=1}^n \int_{S_{n,i}} \left| f(\hat{N}_n^{(i)}(x)) - f(X_i) \right| d\mu(x).
 \end{aligned}$$

Further, since f is Hölder continuous with exponent $s \in (0, 1]$, we have, by the triangle inequality and the sub-additivity of the function $x \mapsto x^s$,

$$\begin{aligned}
 R_n &\leq \frac{L}{n} \sum_{i=1}^n \int_{S_{n,i}} \rho(\hat{N}_n^{(i)}(x), X_i)^s d\mu(x) \\
 &\leq \frac{L}{n} \sum_{i=1}^n \int_{S_{n,i}} \left\{ \rho(\hat{N}_n^{(i)}(x), x) + \rho(x, X_i) \right\}^s d\mu(x) \\
 &\leq \frac{L}{n} \sum_{i=1}^n \int_{S_{n,i}} \left\{ \rho(\hat{N}_n^{(i)}(x), x)^s + \rho(x, X_i)^s \right\} d\mu(x) \\
 &= \frac{L}{n} \sum_{i=1}^n \int_{S_{n,i}} \left\{ \hat{\tau}_n^{(i)}(x)^s + \hat{\tau}_n(x)^s \right\} d\mu(x).
 \end{aligned}$$

Concerning R_n^2 , one can use Jensen's inequality to obtain

$$\begin{aligned}
 R_n^2 &= \left| \frac{1}{n} \sum_{i=1}^n \left\{ \mu(\hat{f}_n^{(i)}) - \mu(\hat{f}_n) \right\} \right|^2 \\
 &\leq \frac{1}{n} \sum_{i=1}^n \left| \mu(\hat{f}_n^{(i)}) - \mu(\hat{f}_n) \right|^2 \\
 &\leq \frac{L^2}{n} \sum_{i=1}^n \int_{S_{n,i}} \left\{ \rho(\hat{N}_n^{(i)}(x), x)^s + \rho(x, X_i)^s \right\}^2 d\mu(x) \\
 &\leq \frac{2L^2}{n} \sum_{i=1}^n \int_{S_{n,i}} \left\{ \rho(\hat{N}_n^{(i)}(x), x)^{2s} + \rho(x, X_i)^{2s} \right\} d\mu(x).
 \end{aligned}$$

For $x \in S_{n,i}$, the nearest neighbor in $\{X_1, \dots, X_n\}$ is X_i , and thus

$$\hat{\tau}_n^{(i)}(x) = \hat{\tau}_{n,2}(x)$$

is the distance to the second nearest neighbor in $\{X_1, \dots, X_n\}$. We get

$$\begin{aligned} R_n &\leq \frac{L}{n} \sum_{i=1}^n \int_{S_{n,i}} \{\hat{\tau}_{n,2}(x)^s + \hat{\tau}_n(x)^s\} d\mu(x) \\ &= \frac{L}{n} \int_M \{\hat{\tau}_{n,2}(x)^s + \hat{\tau}_n(x)^s\} d\mu(x) \end{aligned}$$

and, similarly,

$$\begin{aligned} R_n^2 &\leq \frac{2L^2}{n} \sum_{i=1}^n \int_{S_{n,i}} \{\hat{\tau}_{n,2}(x)^{2s} + \hat{\tau}_n(x)^{2s}\} d\mu(x) \\ &= \frac{2L^2}{n} \int_M \{\hat{\tau}_{n,2}(x)^{2s} + \hat{\tau}_n(x)^{2s}\} d\mu(x). \end{aligned}$$

Consequently, by Lemma 3.3,

$$\begin{aligned} E[R_n] &\leq \frac{L}{n} \int_M E[\hat{\tau}_{n,2}(x)^s + \hat{\tau}_n(x)^s] d\mu(x) \\ &\leq \frac{L}{n} \left(\sup_{x \in M} E[\hat{\tau}_{n,2}(x)^s] + \sup_{x \in M} E[\hat{\tau}_n(x)^s] \right) \\ &\leq \frac{L}{n} \left\{ (C_0 n)^{-s/d} \Gamma(s/d + 2) + (C_0 n)^{-s/d} \Gamma(s/d + 1) \right\} \\ &= (s/d + 2) L C_0^{-s/d} \Gamma(s/d + 1) n^{-1-s/d} \end{aligned}$$

and

$$\begin{aligned} E[R_n^2] &\leq \frac{2L^2}{n} \int_M E[\hat{\tau}_{n,2}(x)^{2s} + \hat{\tau}_n(x)^{2s}] d\mu(x) \\ &\leq \frac{2L^2}{n} \left(\sup_{x \in M} E[\hat{\tau}_{n,2}(x)^{2s}] + \sup_{x \in M} E[\hat{\tau}_n(x)^{2s}] \right) \\ &\leq \frac{2L^2}{n} \left\{ (C_0 n)^{-2s/d} \Gamma(2s/d + 2) + (C_0 n)^{-2s/d} \Gamma(2s/d + 1) \right\} \\ &= 4(s/d + 1) L^2 C_0^{-2s/d} \Gamma(2s/d + 1) n^{-1-2s/d}. \end{aligned}$$

3.6.4 Proofs of Theorems

Proof of Theorem 3.1 for $\hat{\mu}_n^{(\text{NN-loo})}$

Let $Y_{n,i} = \hat{f}_n^{(i)}(X_i) - \mu(\hat{f}_n^{(i)})$ and write

$$\hat{\mu}_n^{(\text{NN-loo})}(f) - \mu(f) = \frac{1}{n} \sum_{i=1}^n (Y_i - Y_{n,i})$$

with $Y_i = f(X_i) - \mu(f)$. Then write

$$\begin{aligned}
 & n^2 \mathbb{E} \left[\left\{ \hat{\mu}_n^{(\text{NN-loo})}(f) - \mu(f) \right\}^2 \right] \\
 &= \sum_{i=1}^n \mathbb{E} [(Y_i - Y_{n,i})^2] + \sum_{i \neq j} \mathbb{E} [(Y_i - Y_{n,i}) (Y_j - Y_{n,j})] \\
 &= n \mathbb{E} [(Y_1 - Y_{n,1})^2] + n(n-1) \mathbb{E} [(Y_1 - Y_{n,1}) (Y_2 - Y_{n,2})]. \quad (3.11)
 \end{aligned}$$

We decompose $Y_{n,1}$ into two terms, one of which does not depend on X_2 . We also use the fact that the Voronoi partition made with $(n-1)$ points is more detailed than the one constructed with $(n-2)$ points, i.e., $S_{n-1,i}^{(1)} \subset S_{n-2,i}^{(1,2)}$ for $i = 3, \dots, n$. Define the map $\mathcal{N}^{(1,2)} : M \rightarrow M$ such that $\mathcal{N}^{(1,2)}(x)$ is the nearest neighbor to x among the sample $\{X_3, X_4, \dots, X_n\}$. Using the identity $\mathcal{N}^{(1,2)}(x) = X_i$ whenever $x \in S_{n-1,i}^{(1)}$ for $i \geq 3$, we write

$$\begin{aligned}
 \hat{f}_n^{(1,2)}(x) &= f(\mathcal{N}^{(1,2)}(x)) \\
 &= f(\mathcal{N}^{(1,2)}(x)) \left(\sum_{i=2}^n \mathbb{1}_{S_{n-1,i}^{(1)}}(x) \right) \\
 &= f(\mathcal{N}^{(1,2)}(x)) \mathbb{1}_{S_{n-1,2}^{(1)}}(x) + \sum_{i=3}^n f(X_i) \mathbb{1}_{S_{n-1,i}^{(1)}}(x) \\
 &= \left\{ f(\mathcal{N}^{(1,2)}(x)) - f(X_2) \right\} \mathbb{1}_{S_{n-1,2}^{(1)}}(x) + \sum_{i=2}^n f(X_i) \mathbb{1}_{S_{n-1,i}^{(1)}}(x).
 \end{aligned}$$

It follows that

$$\hat{f}_n^{(1)}(x) = \hat{L}^{(1)}(x) + \hat{f}_n^{(1,2)}(x)$$

with $\hat{L}^{(1)}(x) = \{f(X_2) - f(\mathcal{N}^{(1,2)}(x))\} \mathbb{1}_{S_{n-1,2}^{(1)}}(x)$. Therefore, since $Y_{n,1} = \hat{f}_n^{(1)}(X_1) - \mu(\hat{f}_n^{(1)})$, we get

$$Y_1 - Y_{n,1} = Y_1 - \left\{ \hat{L}^{(1)}(X_1) - \mu(\hat{L}^{(1)}) \right\} - \left\{ \hat{f}_n^{(1,2)}(X_1) - \mu(\hat{f}_n^{(1,2)}) \right\}.$$

Write

$$\begin{aligned}
 A_1 &= Y_1 = f(X_1) - \mu(f), \\
 A_2 &= Y_2 = f(X_2) - \mu(f), \\
 B_1 &= \hat{L}^{(1)}(X_1) - \mu(\hat{L}^{(1)}), \\
 B_2 &= \hat{L}^{(2)}(X_2) - \mu(\hat{L}^{(2)}), \\
 C_1 &= \hat{f}_n^{(1,2)}(X_1) - \mu(\hat{f}_n^{(1,2)}), \\
 C_2 &= \hat{f}_n^{(1,2)}(X_2) - \mu(\hat{f}_n^{(1,2)}),
 \end{aligned}$$

where $\hat{L}^{(2)}(x) = \{f(X_1) - f(\mathcal{N}^{(1,2)}(x))\} \mathbb{1}_{S_{n-1,1}^{(2)}}(x)$. Then

$$\begin{aligned} E[(Y_1 - Y_{n,1})(Y_2 - Y_{n,2})] &= E[(A_1 - B_1 - C_1)(A_2 - B_2 - C_2)] \\ &= E[A_1A_2] - E[A_1B_2] - E[A_1C_2] \\ &\quad - E[B_1A_2] + E[B_1B_2] + E[B_1C_2] \\ &\quad - E[C_1A_2] + E[C_1B_2] + E[C_1C_2]. \end{aligned}$$

Since A_1 and A_2 are independent, $E[A_1A_2] = 0$. This also applies to $E[A_1C_2]$ and $E[C_1A_2]$. Considering $E[A_1B_2]$ gives

$$\begin{aligned} E[A_1B_2] &= E\left[Y_1 \left\{\hat{L}^{(2)}(X_2) - \mu(\hat{L}^{(2)})\right\}\right] \\ &= E\left[E\left[Y_1 \left\{\hat{L}^{(2)}(X_2) - \mu(\hat{L}^{(2)})\right\} \mid X_1, X_3, \dots, X_n\right]\right] \\ &= E\left[Y_1 E\left[\hat{L}^{(2)}(X_2) - \mu(\hat{L}^{(2)}) \mid X_1, X_3, \dots, X_n\right]\right] = 0. \end{aligned}$$

Due to similar reasoning, $E[B_1A_2] = 0$, $E[B_1C_2] = 0$ and $E[C_1B_2] = 0$. For $E[C_1C_2]$, we have

$$\begin{aligned} E[C_1C_2] &= E\left[\left\{\hat{f}_n^{(1,2)}(X_1) - \mu(\hat{f}_n^{(1,2)})\right\} \left\{\hat{f}_n^{(1,2)}(X_2) - \mu(\hat{f}_n^{(1,2)})\right\}\right] \\ &= E\left[E\left[\left\{\hat{f}_n^{(1,2)}(X_1) - \mu(\hat{f}_n^{(1,2)})\right\} \left\{\hat{f}_n^{(1,2)}(X_2) - \mu(\hat{f}_n^{(1,2)})\right\} \mid X_3, \dots, X_n\right]\right] \\ &= 0. \end{aligned}$$

Therefore, we get

$$\begin{aligned} E[(Y_1 - Y_{n,1})(Y_2 - Y_{n,2})] &= E[B_1B_2] \\ &= E\left[\left\{\hat{L}^{(1)}(X_1) - \mu(\hat{L}^{(1)})\right\} \left\{\hat{L}^{(2)}(X_2) - \mu(\hat{L}^{(2)})\right\}\right]. \end{aligned}$$

Write the four terms on the right-hand side as A, B, C, D , respectively. The Cauchy-Schwarz inequality gives $\|(A - B)(C - D)\|_1 \leq \|A - B\|_2 \|C - D\|_2$, while the fact that B and D are conditional expectations of A and C , respectively, leads to $\|(A - B)(C - D)\|_1 \leq \|A\|_2 \|C\|_2 = \|A\|_2^2$. As a result,

$$E[(Y_1 - Y_{n,1})(Y_2 - Y_{n,2})] \leq E[\hat{L}^{(1)}(X_1)^2].$$

Using the Hölder property, we obtain, by the triangle inequality and the sub-additivity of $x \mapsto x^s$,

$$\begin{aligned}
\left| \hat{L}^{(1)}(x) \right| &= \left| f(X_2) - f(\mathcal{N}^{(1,2)}(x)) \right| \mathbb{1}_{S_{n-1,2}^{(1)}}(x) \\
&= \left| f(\mathcal{N}^{(1)}(x)) - f(\mathcal{N}^{(1,2)}(x)) \right| \mathbb{1}_{S_{n-1,2}^{(1)}}(x) \\
&\leq L \rho(\mathcal{N}^{(1)}(x), \mathcal{N}^{(1,2)}(x))^s \mathbb{1}_{S_{n-1,2}^{(1)}}(x) \\
&\leq 2L \rho(x, \mathcal{N}^{(1,2)}(x))^s \mathbb{1}_{S_{n-1,2}^{(1)}}(x) \\
&\leq 2L \rho(x, \mathcal{N}^{(1,2)}(x))^s \mathbb{1}_{B(x, \hat{\tau}^{(1)}(x))}(X_2) \\
&\leq 2L \rho(x, \mathcal{N}^{(1,2)}(x))^s \mathbb{1}_{B(x, \hat{\tau}^{(1,2)}(x))}(X_2).
\end{aligned}$$

Hence

$$\mathbb{E} \left[\left| \hat{L}^{(1)}(x) \right|^2 \mid X_3, \dots, X_n \right] \leq 4L^2 \rho(x, \mathcal{N}^{(1,2)}(x))^{2s} \mu \left(B(x, \hat{\tau}^{(1,2)}(x)) \right).$$

Moreover,

$$\mu \left(B(x, \hat{\tau}^{(1,2)}(x)) \right) \leq C_1 \hat{\tau}^{(1,2)}(x)^d.$$

Using the equality $\hat{\tau}^{(1,2)}(x) = \rho(x, \mathcal{N}^{(1,2)}(x))$, we obtain

$$\mathbb{E} \left[\left| \hat{L}^{(1)}(x) \right|^2 \mid X_3, \dots, X_n \right] \leq 4C_1 L^2 \rho(x, \mathcal{N}^{(1,2)}(x))^{2s+d}.$$

Applying to the term

$$(Y_1 - Y_{n,1})^2 = \left[f(X_1) - \hat{f}_n^{(1)}(X_1) - \left\{ \mu(f) - \mu(\hat{f}_n^{(1)}) \right\} \right]^2$$

the same reasoning as above with $A = C = f(X_1) - \hat{f}_n^{(1)}(X_1)$ and $B = D = \mu(f) - \mu(\hat{f}_n^{(1)})$, we get

$$\begin{aligned}
\mathbb{E} \left[(Y_1 - Y_{n,1})^2 \right] &\leq \mathbb{E} \left[\left\{ f(X_1) - \hat{f}_n^{(1)}(X_1) \right\}^2 \right] \\
&= \mathbb{E} \left[\left\{ f(X_1) - f(\mathcal{N}^{(1)}(X_1)) \right\}^2 \right] \\
&\leq L^2 \mathbb{E} \left[\rho(X_1, \mathcal{N}^{(1)}(X_1))^{2s} \right].
\end{aligned}$$

Combining all this with (3.11) gives

$$\begin{aligned}
&\mathbb{E} \left[\left| \hat{\mu}_n^{(\text{NN-loo})}(f) - \mu(f) \right|^2 \right] \\
&\leq L^2 n^{-1} \mathbb{E} \left[\rho(X_1, \mathcal{N}^{(1)}(X_1))^{2s} \right] + 4C_1 L^2 \mathbb{E} \left[\rho(X_1, \mathcal{N}^{(1,2)}(X_1))^{2s+d} \right] \\
&= L^2 n^{-1} \mathbb{E} [\hat{\tau}_{n-1}(X_1)^{2s}] + 4C_1 L^2 \mathbb{E} [\hat{\tau}_{n-2}(X_1)^{2s+d}],
\end{aligned}$$

where X_1 is understood to be independent of the nearest neighbor distance functions $\hat{\tau}_{n-1}$ (based on $X_2, \dots, X_n$) and $\hat{\tau}_{n-2}$ (based on $X_3, \dots, X_n$). Applying Lemma 3.3 to $E[\hat{\tau}_{n-1}(X_1)^{2s} \mid X_1]$ and to $E[\hat{\tau}_{n-2}(X_1)^{2s+d} \mid X_1]$, we get

$$\begin{aligned} E[\hat{\tau}_{n-1}(X_1)^{2s}] &\leq \{(n-1)C_0\}^{-2s/d} \Gamma(2s/d+1), \\ E[\hat{\tau}_{n-2}(X_1)^{2s+d}] &\leq \{(n-2)C_0\}^{-2s/d-1} \Gamma(2s/d+2). \end{aligned}$$

Therefore,

$$\begin{aligned} E\left[\left|\hat{\mu}_n^{(\text{NN-loo})}(f) - \mu(f)\right|^2\right] &\leq L^2 n^{-1} \{(n-1)C_0\}^{-2s/d} \Gamma(2s/d+1) \\ &\quad + 4C_1 L^2 \{(n-2)C_0\}^{-2s/d-1} \Gamma(2s/d+2) \end{aligned}$$

and thus

$$\begin{aligned} n^{1+2s/d} E\left[\left|\hat{\mu}_n^{(\text{NN-loo})}(f) - \mu(f)\right|^2\right] &\leq L^2 C_0^{-2s/d} \Gamma(2s/d+1) \{n/(n-1)\}^{2s/d} \\ &\quad + 4C_1 L^2 C_0^{-2s/d-1} \Gamma(2s/d+2) \{n/(n-2)\}^{2s/d+1}. \end{aligned}$$

Since $n/(n-2) \leq 2$ (from $n \geq 4$), the upper bound can be simplified to

$$5L^2 C_1 C_0^{-2s/d-1} \Gamma(2s/d+2) 2^{2s/d+1}.$$

Proof of Theorem 3.1 for $\hat{\mu}_n^{(\text{NN})}$

The proof follows from combining the previous part of Theorem 3.1 and the following inequality from Proposition 3.3,

$$E\left[\left|\frac{1}{n} \sum_{i=1}^n \mu(\hat{f}_n^{(i)}) - \mu(\hat{f}_n)\right|^2\right] \leq 4(s/d+1) L^2 C_0^{-2s/d} \Gamma(2s/d+1) n^{-1-2s/d}.$$

By Minkowski's inequality, we have

$$\begin{aligned} &\left(E\left[\left|\hat{\mu}_n^{(\text{NN})}(f) - \mu(f)\right|^2\right]\right)^{1/2} \\ &\leq \left(E\left[\left|\hat{\mu}_n^{(\text{NN})}(f) - \hat{\mu}_n^{(\text{NN-loo})}(f)\right|^2\right]\right)^{1/2} + \left(E\left[\left|\hat{\mu}_n^{(\text{NN-loo})}(f) - \mu(f)\right|^2\right]\right)^{1/2}. \end{aligned}$$

In view of (3.3), $E\left[\left|\hat{\mu}_n^{(\text{NN})}(f) - \hat{\mu}_n^{(\text{NN-loo})}(f)\right|^2\right] = E\left[\left|\frac{1}{n} \sum_{i=1}^n \mu(\hat{f}_n^{(i)}) - \mu(\hat{f}_n)\right|^2\right]$.

Therefore,

$$\begin{aligned} &\left(E\left[\left|\hat{\mu}_n^{(\text{NN})}(f) - \mu(f)\right|^2\right]\right)^{1/2} \\ &\leq \sqrt{L^2 C_0^{-2s/d} \Gamma(2s/d+2) n^{-1-2s/d}} \left(2\sqrt{s/d+1} + 2^{s/d} (C_1/C_0)^{1/2} \sqrt{10}\right). \end{aligned}$$

Proof of Theorem 3.2 for $\hat{\mu}_n^{(\text{NN-loo})}$

We will apply Theorem 3.3, showing the bounded difference property in two parts (Step 2). First, in Step 1, we construct a large-probability event A on which the bounded difference property will hold. In order to bound the gap between $\mathbb{E}[\hat{\mu}_n^{(\text{NN-loo})}(f) \mid A]$ and $\mu(f)$, we rely in Step 3 on the identity $\mathbb{E}[\hat{\mu}_n^{(\text{NN-loo})}(f)] = \mu(f)$ and on Lemma 3.6.

Step 1. Let $\lceil x \rceil$ denote the smallest integer upper bound to $x \in \mathbb{R}$. By Lemma 3.5, there exists an event A with probability $\mathbb{P}(A) \geq 1 - \delta_n$ such that on A , we have, for $k = \lceil 8d \log(12n/\delta_n) \rceil$,

$$\begin{aligned} \sup_{x \in \mathcal{X}} \hat{\tau}_{n,k}(x) &\leq \left(\frac{2k}{nbcV_d} \right)^{1/d} = \left(\frac{2\lceil 8d \log(12n/\delta_n) \rceil}{nbcV_d} \right)^{1/d} \\ &\leq \left(\frac{17d \log(12n/\delta_n)}{nbcV_d} \right)^{1/d} = \bar{\tau}_n. \end{aligned}$$

Step 2. On the event A , we consider separately two terms of $\hat{\mu}_n^{(\text{NN-loo})} = n^{-1} \sum_{i=1}^n \{f(X_i) - \hat{f}_n^{(i)}(X_i) + \mu(\hat{f}_n^{(i)})\}$, namely $n^{-1} \sum_{i=1}^n \{f(X_i) - \hat{f}_n^{(i)}(X_i)\}$ and $n^{-1} \sum_{i=1}^n \mu(\hat{f}_n^{(i)})$, in order to make the bounded differences property (3.8) for $\hat{\mu}_n^{(\text{NN-loo})}$ satisfied. Further, we apply Theorem 3.3.

Step 2a. Fix $\ell \in \{1, \dots, n\}$. In the original set of points $X_1, \dots, X_n$, replace X_ℓ by $\tilde{X}_\ell$. Let $\hat{N}_n^{(i)}(x)$ be the nearest neighbor of x among $X_1, \dots, \tilde{X}_\ell, \dots, X_n$ without X_i and define $\hat{f}_n^{(i)}(x) = f(\hat{N}_n^{(i)}(x))$. Note that $\hat{N}_n^{(\ell)}(x) = \hat{N}_n^{(\ell)}(x)$ and hence $\hat{f}_n^{(\ell)}(x) = \hat{f}_n^{(\ell)}(x)$. Let

$$\phi_1 \equiv \phi_1(X_1, \dots, X_n) = \frac{1}{n} \sum_{i=1}^n \{f(X_i) - \hat{f}_n^{(i)}(X_i)\}. \quad (3.12)$$

Since $\hat{f}_n^{(\ell)}(\tilde{X}_\ell) = \hat{f}_n^{(\ell)}(\tilde{X}_\ell)$, we have

$$\begin{aligned}
D_{\ell,1} &= \phi_1(X_1, \dots, X_\ell, \dots, X_n) - \phi_1(X_1, \dots, \tilde{X}_\ell, \dots, X_n) \\
&= \frac{1}{n} \sum_{\substack{i=1 \\ i \neq \ell}}^n \{f(X_i) - \hat{f}_n^{(i)}(X_i)\} + \frac{1}{n} \{f(X_\ell) - \hat{f}_n^{(\ell)}(X_\ell)\} \\
&\quad - \frac{1}{n} \sum_{\substack{i=1 \\ i \neq \ell}}^n \{f(X_i) - \hat{f}_n^{(i)}(X_i)\} - \frac{1}{n} \{f(\tilde{X}_\ell) - \hat{f}_n^{(\ell)}(\tilde{X}_\ell)\} \\
&= \frac{1}{n} \sum_{\substack{i=1 \\ i \neq \ell}}^n \{\hat{f}_n^{(i)}(X_i) - \hat{f}_n^{(i)}(X_i)\} \\
&\quad + \frac{1}{n} \left[\{f(X_\ell) - f(\tilde{X}_\ell)\} - \{\hat{f}_n^{(\ell)}(X_\ell) - \hat{f}_n^{(\ell)}(\tilde{X}_\ell)\} \right].
\end{aligned}$$

Considering the first term of $D_{\ell,1}$, we have, in view of (A3),

$$\begin{aligned}
&\frac{1}{n} \sum_{\substack{i=1 \\ i \neq \ell}}^n \left| \hat{f}_n^{(i)}(X_i) - \hat{f}_n^{(i)}(X_i) \right| \\
&= \frac{1}{n} \sum_{\substack{i=1 \\ i \neq \ell}}^n \left| f(\hat{N}_n^{(i)}(X_i)) - f(\hat{N}_n^{(i)}(X_i)) \right| \\
&\leq \frac{L}{n} \sum_{\substack{i=1 \\ i \neq \ell}}^n \left\| \hat{N}_n^{(i)}(X_i) - \hat{N}_n^{(i)}(X_i) \right\|^s \\
&= \frac{L}{n} \sum_{\substack{i=1 \\ i \neq \ell}}^n \left\| \hat{N}_n^{(i)}(X_i) - \hat{N}_n^{(i)}(X_i) \right\|^s \mathbb{1}_{\{\hat{N}_n^{(i)}(X_i) = \tilde{X}_\ell \text{ or } \hat{N}_n^{(i)}(X_i) = X_\ell\}}.
\end{aligned}$$

By the triangle inequality and the sub-additivity of $x \mapsto x^s$, we get, in hopefully obvious notation,

$$\begin{aligned}
\left\| \hat{N}_n^{(i)}(X_i) - \hat{N}_n^{(i)}(X_i) \right\|^s &\leq \left\| X_i - \hat{N}_n^{(i)}(X_i) \right\|^s + \left\| X_i - \hat{N}_n^{(i)}(X_i) \right\|^s \\
&\leq \left\| X_i - \hat{N}_n^{(i,\ell)}(X_i) \right\|^s + \left\| X_i - \hat{N}_n^{(i,\ell)}(X_i) \right\|^s \\
&\leq 2 \left\| X_i - \hat{N}_n^{(i,\ell)}(X_i) \right\|^s \leq 2 \sup_{x \in \mathcal{X}} \hat{\tau}_{n,3}(x)^s, \quad (3.13)
\end{aligned}$$

where $\hat{\tau}_{n,3}(x)$ is the distance to the third nearest neighbor of x among $X_1, \dots, X_n$. By Lemma 3.4,

$$\sum_{\substack{i=1 \\ i \neq \ell}}^n \mathbb{1}_{\{\hat{N}_n^{(i)}(X_i) = \tilde{X}_\ell \text{ or } \hat{N}_n^{(i)}(X_i) = X_\ell\}} \leq 2\psi_d. \quad (3.14)$$

Therefore, by (3.13) and (3.14),

$$\frac{1}{n} \sum_{\substack{i=1 \\ i \neq \ell}}^n \left| \hat{f}_n^{(i)}(X_i) - \hat{f}_n^{(i)}(X_i) \right| \leq \frac{4L\psi_d}{n} \sup_{x \in \mathcal{X}} \hat{\tau}_{n,3}(x)^s. \quad (3.15)$$

Considering the second term of $D_{\ell,1}$, we have

$$\begin{aligned} & \frac{1}{n} \left| \{f(X_\ell) - f(\tilde{X}_\ell)\} - \{\hat{f}_n^{(\ell)}(X_\ell) - \hat{f}_n^{(\ell)}(\tilde{X}_\ell)\} \right| \\ & \leq \frac{L}{n} \left[\left\| X_\ell - \hat{N}_n^{(\ell)}(X_\ell) \right\|^s + \left\| \tilde{X}_\ell - \hat{N}_n^{(\ell)}(\tilde{X}_\ell) \right\|^s \right] \\ & \leq \frac{2L}{n} \sup_{x \in \mathcal{X}} \hat{\tau}_{n,3}(x)^s. \end{aligned} \quad (3.16)$$

On the event A , by (3.15) and (3.16), we therefore have, since $k \geq 3$ for our choice of k in Step 1, the bound

$$|D_{\ell,1}| \leq \frac{2L}{n} (2\psi_d + 1) \bar{\tau}_n^s.$$

Step 2b. Let $\phi_2 \equiv \phi_2(X_1, \dots, X_n) = n^{-1} \sum_{i=1}^n \mu(\hat{f}_n^{(i)})$. Therefore,

$$\begin{aligned} D_{\ell,2} &= \phi_2(X_1, \dots, X_\ell, \dots, X_n) - \phi_2(X_1, \dots, \tilde{X}_\ell, \dots, X_n) \\ &= \frac{1}{n} \sum_{i=1}^n \{\mu(\hat{f}_n^{(i)}) - \mu(\hat{f}_n^{(i)})\} = \frac{1}{n} \sum_{i=1}^n \mu(\hat{f}_n^{(i)} - \hat{f}_n^{(i)}). \end{aligned}$$

Let $S_{n+1,j}$, for $j = 1, \dots, n+1$, be the Voronoi cells induced by $\mathcal{X}_{n+1} = \{X_1, \dots, X_n, \tilde{X}_\ell\}$: for $x \in S_{n+1,j}$ and $j = 1, \dots, n+1$, the nearest neighbor of x among $\mathcal{X}_{n+1}$ is X_j , where $X_{n+1} := \tilde{X}_\ell$. Clearly,

$$\forall j \in \{1, \dots, n+1\} \setminus \{i, \ell, n+1\}, \quad \forall x \in S_{n+1,j}, \quad \hat{f}_n^{(i)}(x) = f(X_j) = \hat{f}_n^{(i)}(x).$$

It follows that

$$D_{\ell,2} = \frac{1}{n} \sum_{i=1}^n \mu \left((\hat{f}_n^{(i)} - \hat{f}_n^{(i)}) \mathbb{1}_{S_{n+1,i} \cup S_{n+1,\ell} \cup S_{n+1,n+1}} \right).$$

Moreover, by the triangle inequality, the sub-additivity of $x \mapsto x^s$ and (A3), we have, for $x \in S_{n+1,i} \cup S_{n+1,\ell} \cup S_{n+1,n+1}$,

$$\begin{aligned} \left| \hat{f}_n^{(i)}(x) - \hat{f}_n^{(i)}(x) \right| &\leq \left| \hat{f}_n^{(i)}(x) - f(x) \right| + \left| \hat{f}_n^{(i)}(x) - f(x) \right| \\ &\leq L \cdot \left\| \hat{N}_n^{(i)}(x) - x \right\|^s + L \cdot \left\| \hat{N}_n^{(i)}(x) - x \right\|^s \\ &\leq 2L \sup_{x \in \mathcal{X}} \hat{\tau}_{n,3}(x)^s. \end{aligned}$$

Clearly, $\sum_{i=1}^n \mu(S_{n+1,i}) = \mu(\bigcup_{i=1}^n S_{n+1,i}) \leq 1$. On the event A , we thus obtain

$$\begin{aligned} |D_{\ell,2}| &\leq \frac{1}{n} \sum_{i=1}^n \mu \left(\left| \hat{f}_n^{(i)} - \hat{\tilde{f}}_n^{(i)} \right| \mathbb{1}_{S_{n+1,i} \cup S_{n+1,\ell} \cup S_{n+1,n+1}} \right) \\ &\leq 2L\bar{\tau}_n^s \cdot \frac{1}{n} \sum_{i=1}^n \mu(S_{n+1,i} \cup S_{n+1,\ell} \cup S_{n+1,n+1}) \\ &\leq 2L\bar{\tau}_n^s \cdot \{n^{-1} + \mu(S_{n+1,\ell}) + \mu(S_{n+1,n+1})\}. \end{aligned}$$

Recall that λ_d denotes the d -dimensional Lebesgue-measure and that the density, p , of μ satisfies $\sup_{x \in M} p(x) \leq U$. For any $j = 1, \dots, n+1$, the μ -volume of a Voronoi cell satisfies

$$\mu(S_{n+1,j}) = \int_{S_{n+1,j}} p \, d\lambda_d \leq U \cdot \lambda_d(S_{n+1,j} \cap M).$$

Letting $\hat{\tau}_{n+1,1}(x)$ denote the nearest neighbor of $x \in M$ in $\mathcal{X}_{n+1}$, we have

$$\forall x \in S_{n+1,j} \cap M, \quad \|x - X_j\| = \hat{\tau}_{n+1,1}(x) \leq \sup_{x' \in M} \hat{\tau}_{n+1,1}(x').$$

On the event A , the latter is bounded by $\bar{\tau}_n$, which implies that $S_{n+1,j} \cap M$ is contained in a ball of radius $\bar{\tau}_n$ centered at X_j . We find

$$\max_{j=1,\dots,n+1} \mu(S_{n+1,j}) \leq U \cdot \bar{\tau}_n^d V_d, \quad (3.17)$$

with V_d the volume of the unit ball in $\mathbb{R}^d$. We conclude that, on the event A ,

$$|D_{\ell,2}| \leq 2L\bar{\tau}_n^s \cdot \left(n^{-1} + 2UV_d\bar{\tau}_n^d \right).$$

Step 2c. Let $\phi_3 = \phi_1 + \phi_2 = n^{-1} \sum_{i=1}^n \{f(X_i) - \hat{f}_n^{(i)}(X_i) + \mu(\hat{f}_n^{(i)})\} = \hat{\mu}_n^{(\text{NN-loo})}$. Then we get, with probability at least $1 - \delta_n$,

$$\begin{aligned} D_{\ell,3} &= |\phi_3(X_1, \dots, X_\ell, \dots, X_n) - \phi_3(X_1, \dots, \tilde{X}_\ell, \dots, X_n)| \leq |D_{\ell,1}| + |D_{\ell,2}| \\ &\leq 2Ln^{-1}(2\psi_d + 1)\bar{\tau}_n^s + 2L\bar{\tau}_n^s \cdot \left(n^{-1} + 2UV_d\bar{\tau}_n^d \right) \\ &= 2Ln^{-1}(2\psi_d + 2)\bar{\tau}_n^s + 4LUV_d\bar{\tau}_n^{s+d} \\ &= 4L(\psi_d + 1)n^{-1} \left(\frac{17d \log(12n/\delta_n)}{nbcV_d} \right)^{s/d} + 4LUV_d \left(\frac{17d \log(12n/\delta_n)}{nbcV_d} \right)^{1+s/d} \\ &\leq K_1 \left(\frac{\log(12n/\delta_n)}{n} \right)^{1+s/d} = c_{\ell,3} \end{aligned}$$

with $K_1 = 4L \left(\frac{17d}{bcV_d} \right)^{s/d} \left(\psi_d + 1 + \frac{17dU}{bc} \right)$.

We have

$$\begin{aligned}\sum_{\ell=1}^n c_{\ell,3} &= nc_{\ell,3} = K_1 n^{-s/d} (\log(12n/\delta_n))^{1+s/d}, \\ \sum_{\ell=1}^n c_{\ell,3}^2 &= nc_{\ell,3}^2 = K_1^2 n^{-1-2s/d} (\log(12n/\delta_n))^{2+2s/d}.\end{aligned}$$

We apply Theorem 3.3 to $\phi_3 = \phi_3(X_1, \dots, X_n) = n^{-1} \sum_{i=1}^n \{f(X_i) - \hat{f}_n^{(i)}(X_i) + \mu(\hat{f}_n^{(i)})\}$. For $t \geq \delta_n \sum_{\ell=1}^n c_{\ell,3} = \delta_n K_1 n^{-s/d} (\log(12n/\delta_n))^{1+s/d}$, we get

$$\begin{aligned}P(|\phi_3 - E(\phi_3 | A)| \geq t) \\ \leq \delta_n + 2 \exp\left(-\frac{2 \left\{t - \delta_n K_1 n^{-s/d} (\log(12n/\delta_n))^{1+s/d}\right\}^2}{K_1^2 n^{-1-2s/d} (\log(12n/\delta_n))^{2+2s/d}}\right).\end{aligned}\quad (3.18)$$

Let $0 < \delta < 1 - \delta_n$. Then

$$P(|\phi_3 - E(\phi_3 | A)| \geq t) \leq \delta + \delta_n$$

provided t is such that the exponential function in (3.18) is bounded by $\delta/2$, which happens if

$$\begin{aligned}t &\geq \sqrt{\frac{K_1^2 \log(2/\delta) (\log(12n/\delta_n))^{2+2s/d}}{2n^{1+2s/d}}} + \frac{K_1 \delta_n (\log(12n/\delta_n))^{1+s/d}}{n^{s/d}} \\ &= K_1 \left(n^{-1/2} \sqrt{\log(2/\delta)/2} + \delta_n\right) \frac{(\log(12n/\delta_n))^{1+s/d}}{n^{s/d}}.\end{aligned}$$

Hence, for $0 < \delta \leq 1 - \delta_n$, with probability at least $1 - (\delta + \delta_n)$, we have

$$|\phi_3 - E(\phi_3 | A)| < K_1 \left(n^{-1/2} \sqrt{\log(2/\delta)/2} + \delta_n\right) \frac{(\log(12n/\delta_n))^{1+s/d}}{n^{s/d}}.$$

Step 3. We have $\|\phi_3\|_\infty \leq 3 \|f\|_\infty$. By Lemma 3.6,

$$|E(\phi_3) - E(\phi_3 | A)| \leq 2\delta_n \|\phi_3\|_\infty \leq 6\delta_n \|f\|_\infty.$$

Thus, we obtain, with probability at least $1 - (\delta + \delta_n)$, for $0 < \delta \leq 1 - \delta_n$,

$$\begin{aligned}\left|\hat{\mu}_n^{(\text{NN-loo})}(f) - \mu(f)\right| \\ &= |\phi_3 - E(\phi_3)| \\ &\leq |\phi_3 - E(\phi_3 | A)| + |E(\phi_3 | A) - E(\phi_3)| \\ &\leq K_1 \left(n^{-1/2} \sqrt{\log(2/\delta)/2} + \delta_n\right) \frac{(\log(12n/\delta_n))^{1+s/d}}{n^{s/d}} + 6\delta_n \|f\|_\infty.\end{aligned}$$

Step 4. Fix $\varepsilon > 0$ and choose $\delta_n = \min(\varepsilon/2, n^{-1/2-s/d})$, $\delta = \varepsilon - \delta_n$. Then $\delta = \varepsilon - \min(\varepsilon/2, n^{-1/2-s/d}) = \max(\varepsilon/2, \varepsilon - n^{-1/2-s/d})$. Considering the case when $\varepsilon \leq \frac{2}{n^{1/2+s/d}}$, we get

$$\begin{aligned}
& \left| \hat{\mu}_n^{(\text{NN-loo})}(f) - \mu(f) \right| \\
& \leq K_1 \left(\frac{\sqrt{\log(4/\varepsilon)/2}}{n^{1/2}} + \frac{\varepsilon}{2} \right) \frac{(\log(24n/\varepsilon))^{1+s/d}}{n^{s/d}} + 3\varepsilon \|f\|_\infty \\
& \leq K_1 \left(\frac{\sqrt{\log(4/\varepsilon)/2}}{n^{1/2}} + \frac{1}{n^{1/2+s/d}} \right) \frac{(\log(24n/\varepsilon))^{1+s/d}}{n^{s/d}} + \frac{6\|f\|_\infty}{n^{1/2+s/d}} \\
& \leq K_1 \left(\sqrt{\log\left(\frac{4}{\varepsilon}\right)/2} + \frac{1}{n^{s/d}} \right) \frac{(\log(24n/\varepsilon))^{1+s/d}}{n^{1/2+s/d}} + \frac{6\|f\|_\infty}{n^{1/2+s/d}}.
\end{aligned}$$

Otherwise, when $\varepsilon > \frac{2}{n^{1/2+s/d}}$, using that $\log(12n^{3/2+s/d}) \leq \log((3n)^3)$, we have

$$\begin{aligned}
& \left| \hat{\mu}_n^{(\text{NN-loo})}(f) - \mu(f) \right| \\
& \leq K_1 \left(\sqrt{\log\left(\frac{2}{\varepsilon - n^{-1/2-s/d}}\right)/2} + \frac{1}{n^{s/d}} \right) \frac{(\log(12n^{3/2+s/d}))^{1+s/d}}{n^{1/2+s/d}} \\
& \quad + \frac{6\|f\|_\infty}{n^{1/2+s/d}} \\
& \leq K_1 \left(\sqrt{\log\left(\frac{4}{\varepsilon}\right)/2} + \frac{1}{n^{s/d}} \right) \frac{(3\log(3n))^{1+s/d}}{n^{1/2+s/d}} + \frac{6\|f\|_\infty}{n^{1/2+s/d}}.
\end{aligned}$$

Therefore, for any $\varepsilon \in (0, 1)$, we have, with probability at least $1 - \varepsilon$,

$$\begin{aligned}
& \left| \hat{\mu}_n^{(\text{NN-loo})}(f) - \mu(f) \right| \leq \frac{6\|f\|_\infty}{n^{1/2+s/d}} \\
& + K_1 \left(\sqrt{\log\left(\frac{4}{\varepsilon}\right)/2} + \frac{1}{n^{s/d}} \right) \times \begin{cases} \frac{(\log(24n/\varepsilon))^{1+s/d}}{n^{1/2+s/d}}, & \text{for } \varepsilon \leq \frac{2}{n^{1/2+s/d}}, \\ \frac{(3\log(3n))^{1+s/d}}{n^{1/2+s/d}}, & \text{for } \varepsilon > \frac{2}{n^{1/2+s/d}}. \end{cases}
\end{aligned}$$

Since $\sum_{i=1}^n w_{n,i} = 1$, we can replace f by $f - C$ for any constant C and thus, choosing C optimally, replace $\|f\|_\infty$ by half the diameter of the range of f , that is, by $C_f = \frac{1}{2} \{ \sup_{x \in M} f(x) - \inf_{x \in M} f(x) \}$.

Proof of Theorem 3.2 for $\hat{\mu}_n^{(\text{NN})}$

Consider the same event A and the same upper bound $\bar{\tau}_n$ as in Step 1 in the proof of Theorem 3.2 for $\hat{\mu}_n^{(\text{NN-loo})}$. We write

$$\hat{\mu}_n^{(\text{NN})}(f) = \phi_5 \equiv \phi_5(X_1, \dots, X_n) = \phi_1(X_1, \dots, X_n) + \phi_4(X_1, \dots, X_n)$$

with ϕ_1 as in Eq. (3.12) and with

$$\phi_4 \equiv \phi_4(X_1, \dots, X_n) = \mu(\hat{f}_n).$$

Step 1. Let $\hat{N}_n(x)$ be the nearest neighbor of x among $X_1, \dots, \tilde{X}_\ell, \dots, X_n$ and let $\hat{f}_n(x) = f(\hat{N}_n(x))$. Then

$$\begin{aligned} D_{\ell,4} &= \phi_4(X_1, \dots, X_\ell, \dots, X_n) - \phi_4(X_1, \dots, \tilde{X}_\ell, \dots, X_n) \\ &= \mu(\hat{f}_n) - \mu(\tilde{f}_n) = \mu(\hat{f}_n - \tilde{f}_n). \end{aligned}$$

For $i = 1, \dots, n+1$, define $S_{n+1,i}$ as in Step 2b in the proof of Theorem 3.2 for $\hat{\mu}_n^{(\text{NN-loo})}$. It follows that

$$D_{\ell,4} = \mu \left((\hat{f}_n - \tilde{f}_n) \mathbb{1}_{S_{n+1,\ell} \cup S_{n+1,n+1}} \right).$$

Further, by the triangle inequality, the sub-additivity of the function $x \mapsto x^s$ and the Hölder- s property, we have, for $x \in M \cap (S_{n+1,\ell} \cup S_{n+1,n+1})$,

$$\begin{aligned} \left| \hat{f}_n(x) - \tilde{f}_n(x) \right| &\leq \left| \hat{f}_n(x) - f(x) \right| + \left| \tilde{f}_n - f(x) \right| \\ &\leq L \cdot \left\| \hat{N}_n(x) - x \right\|^s + L \cdot \left\| \hat{N}_n(x) - x \right\|^s \\ &\leq 2L \sup_{x' \in M} \hat{\tau}_{n,2}(x')^s. \end{aligned}$$

On the event A , we thus obtain

$$\left| D_{\ell,4} \right| \leq 2L\bar{\tau}_n^s \cdot \mu(S_{n+1,\ell} \cup S_{n+1,n+1}) \leq 2L\bar{\tau}_n^s \cdot \{ \mu(S_{n+1,\ell}) + \mu(S_{n+1,n+1}) \}.$$

By Eq. (3.17) in Step 2b in the proof of Theorem 3.2 for $\hat{\mu}_n^{(\text{NN-loo})}$, we have, on the event A ,

$$\max_{j=1, \dots, n+1} \mu(S_{n+1,j}) \leq U \cdot \bar{\tau}_n^d V_d,$$

with V_d the volume of the unit ball in $\mathbb{R}^d$. Therefore, we conclude that, on the event A , we have

$$\left| D_{\ell,4} \right| \leq 4LUV_d \bar{\tau}_n^{s+d}. \quad (3.19)$$

Step 2. By (3.19) and by the result of Step 2a in the proof of Theorem 3.2 for $\hat{\mu}_n^{(\text{NN-loo})}$, we have

$$\begin{aligned}
D_{\ell,5} &= |\phi_5(X_1, \dots, X_\ell, \dots, X_n) - \phi_5(X_1, \dots, \tilde{X}_\ell, \dots, X_n)| \\
&\leq |D_{\ell,1}| + |D_{\ell,4}| \\
&\leq 2Ln^{-1}(2\psi_d + 1)\bar{\tau}_n^s + 4LUV_d\bar{\tau}_n^{s+d} \\
&= 2Ln^{-1}(2\psi_d + 1) \left(\frac{17d \log(12n/\delta_n)}{nbcV_d} \right)^{s/d} + 4LUV_d \left(\frac{17d \log(12n/\delta_n)}{nbcV_d} \right)^{1+s/d} \\
&\leq K_2 \left(\frac{\log(12n/\delta_n)}{n} \right)^{1+s/d} = c_{\ell,5}
\end{aligned}$$

with $K_2 = 2L \left(\frac{17d}{bcV_d} \right)^{s/d} \left(2\psi_d + 1 + \frac{34dU}{bc} \right)$.

We have

$$\begin{aligned}
\sum_{\ell=1}^n c_{\ell,5} &= nc_{\ell,5} = K_2 n^{-s/d} (\log(12n/\delta_n))^{1+s/d}, \\
\sum_{\ell=1}^n c_{\ell,5}^2 &= nc_{\ell,5}^2 = K_2^2 n^{-1-2s/d} (\log(12n/\delta_n))^{2+2s/d}.
\end{aligned}$$

Thanks to Theorem 3.3, it follows that, for $t \geq \delta_n \sum_{\ell=1}^n c_{\ell,5} = \delta_n K_2 n^{-s/d} (\log(12n/\delta_n))^{1+s/d}$,

$$\begin{aligned}
&\mathbb{P}(|\phi_5 - \mathbb{E}(\phi_5 | A)| \geq t) \\
&\leq \delta_n + 2 \exp \left(- \frac{2 \{t - \delta_n K_2 n^{-s/d} (\log(12n/\delta_n))^{1+s/d}\}^2}{K_2^2 n^{-1-2s/d} (\log(12n/\delta_n))^{2+2s/d}} \right).
\end{aligned}$$

Let $0 < \delta \leq 1 - \delta_n$. Then

$$\mathbb{P}(|\phi_5 - \mathbb{E}(\phi_5 | A)| \geq t) \leq \delta + \delta_n$$

provided

$$t \geq K_2 \left(n^{-1/2} \sqrt{\log(2/\delta)/2} + \delta_n \right) \frac{(\log(12n/\delta_n))^{1+s/d}}{n^{s/d}}.$$

Hence, for $0 < \delta < 1 - \delta_n$, with probability at least $1 - (\delta + \delta_n)$, we have

$$|\phi_5 - \mathbb{E}(\phi_5 | A)| < K_2 \left(n^{-1/2} \sqrt{\log(2/\delta)/2} + \delta_n \right) \frac{(\log(12n/\delta_n))^{1+s/d}}{n^{s/d}}. \quad (3.20)$$

Step 3. Since $\|\phi_5\|_\infty \leq 3\|f\|_\infty$, we have, by Lemma 3.6,

$$|\mathbb{E}(\phi_5 | A) - \mathbb{E}(\phi_5)| \leq 2\delta_n \|\phi_5\|_\infty \leq 6\delta_n \|f\|_\infty.$$

Recall $E[\hat{\mu}_n^{(\text{NN-loo})}(f)] = \mu(f)$. On the event on which (3.20) holds, we have, in view of Proposition 3.3 with $C_0 = V_d bc$, for $n \geq 4$,

$$\begin{aligned}
& \left| \hat{\mu}_n^{(\text{NN})}(f) - \mu(f) \right| \\
& \leq |\phi_5 - E(\phi_5 | A)| + |E(\phi_5 | A) - E(\phi_5)| \\
& \quad + \left| E \left[\hat{\mu}_n^{(\text{NN})}(f) \right] - E \left[\hat{\mu}_n^{(\text{NN-loo})}(f) \right] \right| \\
& \leq K_2 \left(n^{-1/2} \sqrt{\log(2/\delta)/2} + \delta_n \right) \frac{(\log(12n/\delta_n))^{1+s/d}}{n^{s/d}} + 6\delta_n \|f\|_\infty \\
& \quad + \frac{(s/d+2)L(V_d bc)^{-s/d}\Gamma(s/d+1)}{n^{1+s/d}} \\
& \leq K_2 \left(n^{-1/2} \sqrt{\log(2/\delta)/2} + \delta_n \right) \frac{(\log(12n/\delta_n))^{1+s/d}}{n^{s/d}} + 6\delta_n \|f\|_\infty \\
& \quad + \frac{(s/d+2)L(V_d bc)^{-s/d}}{n^{1+s/d}}.
\end{aligned}$$

Step 4. Similarly as in Step 4 of the proof of Theorem 3.2 for $\hat{\mu}_n^{(\text{NN-loo})}$, fix $\varepsilon > 0$ and choose $\delta_n = \min(\varepsilon/2, n^{-1/2-1/d})$, $\delta = \varepsilon - \delta_n$. Then, for any $\varepsilon \in (0, 1)$, we have, with probability at least $1 - \varepsilon$,

$$\begin{aligned}
& \left| \hat{\mu}_n^{(\text{NN})}(f) - \mu(f) \right| \leq \frac{6\|f\|_\infty}{n^{1/2+s/d}} + \frac{(s/d+2)L(V_d bc)^{-s/d}}{n^{1+s/d}} \\
& \quad + K_2 \left(\sqrt{\frac{\log(4/\varepsilon)}{2}} + \frac{1}{n^{s/d}} \right) \times \begin{cases} \frac{(\log(24n/\varepsilon))^{1+s/d}}{n^{1/2+s/d}}, & \text{for } \varepsilon \leq \frac{2}{n^{1/2+s/d}}, \\ \frac{(3\log(3n))^{1+s/d}}{n^{1/2+s/d}}, & \text{for } \varepsilon > \frac{2}{n^{1/2+s/d}}. \end{cases}
\end{aligned}$$

Proof of Theorem 3.3

We have

$$\begin{aligned}
P(|\phi(X) - m| \geq t) & \leq P(|\phi(X) - m| \geq t, X \in A) + P(X \notin A) \\
& \leq P(\phi(X) - m \geq t, X \in A) + P(-\phi(X) + m \geq t, X \in A) \\
& \quad + p.
\end{aligned}$$

From the proof of Theorem 2.1 in Combes (2015), we get

$$P(\phi(X) - m \geq t, X \in A) \leq \exp \left(-\frac{2 \max(0, t - p\bar{c})^2}{\sum_{\ell=1}^n c_\ell^2} \right).$$

By symmetry,

$$P(-\phi(X) + m \geq t, X \in A) \leq \exp \left(-\frac{2 \max(0, t - p\bar{c})^2}{\sum_{\ell=1}^n c_\ell^2} \right).$$

Thus, we obtain the stated result.

3.6.5 Additional experiments

Finance background. Options are financial derivatives based on the value of underlying securities. They give the buyer the right to buy (call option) or sell (put option) the underlying asset at a pre-determined price within a specific time frame. The price of an option may be expressed as the expectation, under the so-called risk-neutral measure, of the payoff discounted to the present value. Consider a contract of European type, which specifies a payoff $V(S_T)$, depending on the level of the underlying asset S_t at maturity $t = T$. The value V of the contract at time $t = 0$ conditional on an underlying value S_0 is

$$V(S_0) = E_Q[e^{-rT} V(S_T)], \quad (3.21)$$

where E_Q denotes the expectation under the risk-neutral measure and r is the risk-free interest rate. Such a representation suggests a straightforward Monte Carlo based method for its calculation by simulating random paths of the underlying asset, calculating each time the resulting payoff and taking the average of the result. This approach is particularly useful when dealing with exotic options, for which the above expectation often does not permit a closed-form expression.

The payoff of a European call option with strike price K is given by $V(S_T) = (S_T - K)_+$ and depends only on the level of the underlying asset S_t at maturity time $t = T$. In contrast, the payoff of a barrier option (Merton, 1973) depends on the whole path $(S_t)_{t \in [0, T]}$. The option becomes worthless or may be activated upon the crossing of a price point barrier denoted H . More precisely, Knock-Out (KO) options expire worthless when the underlying's spot price crosses the pre-specified barrier level whereas Knock-In (KI) options only come into existence if the pre-specified barrier level is crossed by the underlying asset's price. The payoffs of *up-in* (UI) and *up-out* (UO) barrier options with barrier price K are given by

$$\begin{aligned} V_{(UI)}(S) &= (S_T - K)_+ \int \left\{ \max_{t \in [0, T]} S_t \geq H \right\}, \\ V_{(UO)}(S) &= (S_T - K)_+ \mathbb{1} \left\{ \max_{t \in [0, T]} S_t < H \right\}. \end{aligned} \quad (3.22)$$

Market Dynamics. The Black–Scholes model (Black and Scholes, 1973) is a mathematical model for pricing option contracts. It is based on geometric Brownian motion with constant drift and volatility so that the underlying stock S_t satisfies the following stochastic differential equation:

$$dS_t = \mu S_t dt + \sigma S_t dW_t,$$

where μ represents the drift rate of growth of the underlying stock, σ is the volatility and W denotes a Wiener process. Although simple and widely used in practice, the Black–Scholes model has some limitations. In particular, it assumes constant values for the risk-free rate of return and volatility over the option duration. Neither of those necessarily remains constant in the real world. The Heston model (Heston, 1993) is a type of stochastic volatility model that can be used for pricing options on various securities. For the Heston model, the previous constant volatility σ is replaced by a stochastic volatility v_t which follows an Ornstein–Uhlenbeck process. The underlying stock S_t satisfies the following equations

$$\begin{cases} dS_t = \mu S_t dt + \sqrt{v_t} S_t dW_t^S, \\ dv_t = \kappa(\theta - v_t) dt + \xi \sqrt{v_t} dW_t^v, \end{cases} \quad dW_t^S dW_t^v = \rho dt.$$

with stochastic volatility v_t , drift term μ , long run average variance θ , rate of mean reversion κ and volatility of volatility ξ . Essentially the Heston model is a geometric Brownian motion with non-constant volatility, where the change in S has relationship ρ with the change in volatility.

Monte Carlo procedures. The application of standard Monte Carlo methods to option pricing takes the following form:

- (1) Simulate a large number n of price paths for the underlying asset: $(S_{(1)}, \dots, S_{(n)})$.
- (2) For each path, compute the associated payoff of the option, e.g., as in Eq. (3.22): $(V_1, \dots, V_n)$.
- (3) Average the payoffs and discount them to present value: $\hat{V}_n = (e^{-rT}/n) \sum_{i=1}^n V_i$.

In practice, the price paths are simulated using an Euler scheme with a discretization of the time period $[0, T]$ comprised of m times $t_1 = 0 < t_2 < \dots < t_m = T$. Each price path $S_{(i)}$ for $i = 1, \dots, n$ is actually a vector $(S_{(i)}^{(1)}, \dots, S_{(i)}^{(m)})$, so that the indicator function of the barrier options is computed on the discretized prices. Common values for m are the number of trading days per year which is $m = 252$ for $T = 1$ year.

Parameters. Several numerical experiments are performed for the pricing of European barrier call options “up-in” and “up-out”. The number of sampled paths is $n \in \{500, 1\,000, 2\,000, 3\,000, 5\,000\}$ and the granularity of the grid is equal to $m = 240$. Two different mathematical models are considered when simulating the underlying asset price trajectories:

- (1) the Black–Scholes model with constant volatility $\sigma = 0.30$;

- (2) the Heston model with initial volatility $v_0 = 0.1$, long-run average variance $\theta = 0.02$, rate of mean reversion $\kappa = 4$, instantaneous correlation $\rho = 0.8$ and volatility of volatility $\xi = 0.9$.

In both cases the fixed parameters are: spot price $S_0 = 100$, interest rate $r = 0.10$, maturity $T = 2$ months, strike price $K = S_0 = 100$ and barrier price $H = 130$.

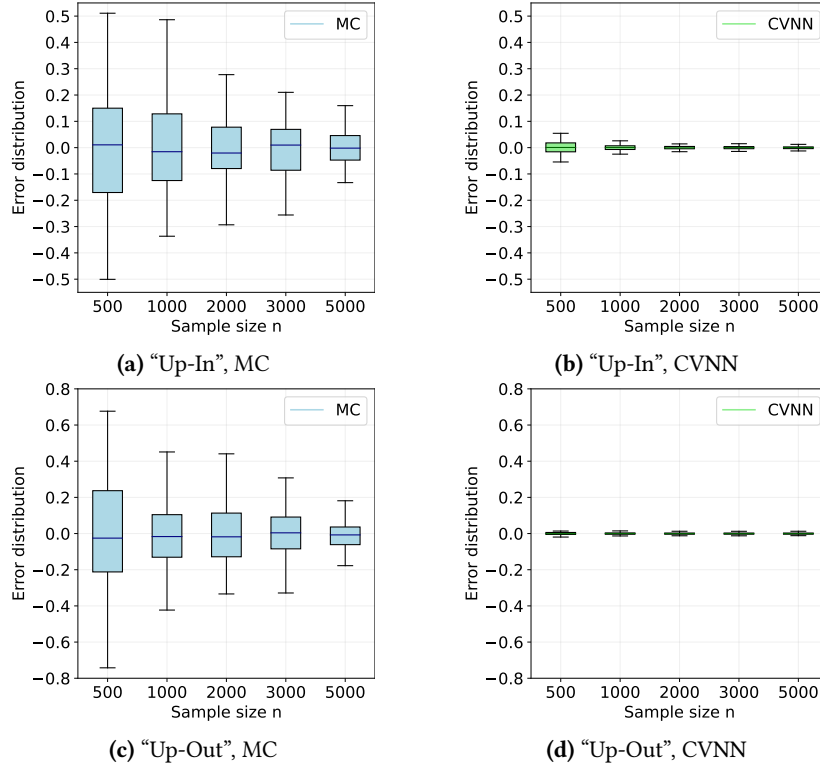


Figure 3.6: Barrier option pricing under Black-Scholes model with spot price $S_0 = 100$, strike $K = S_0$, maturity $T = 2$ months, risk-free rate $r = 0.1$, constant volatility $\sigma = 0.3$, barrier price $H = 130$. The boxplots are obtained over 100 replications.

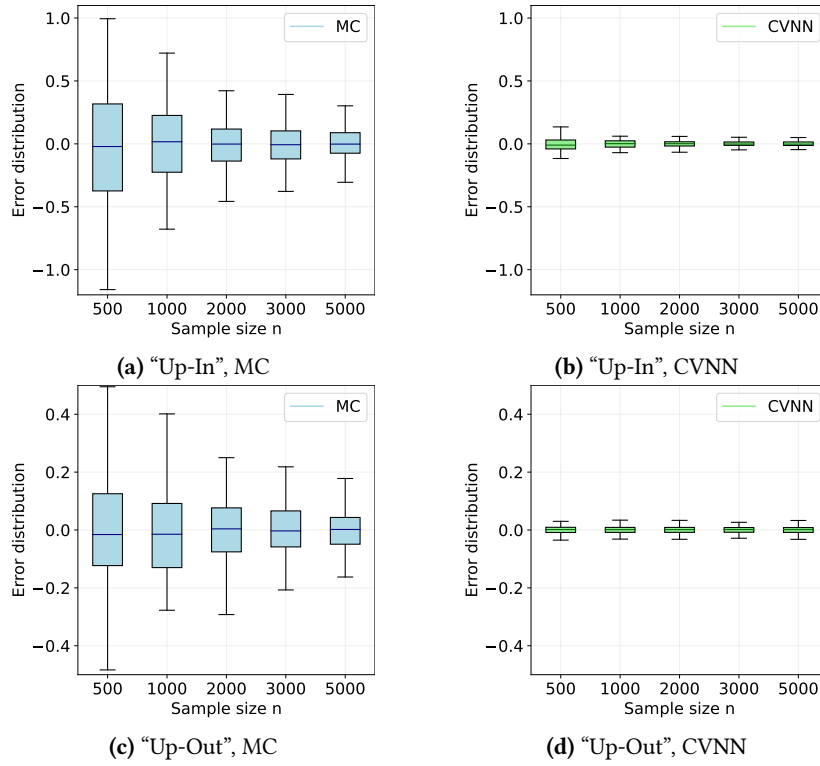


Figure 3.7: Boxplots of barrier option pricing with Heston Model with spot price $S_0 = 100$, strike $K = S_0$, barrier price $H = 130$, maturity $T = 2$ months, risk-free rate $r = 0.1$, initial volatility $v_0 = 0.1$, long-run average variance $\theta = 0.02$, rate of mean reversion $\kappa = 4$, instantaneous correlation $\rho = 0.8$ and volatility of volatility $\xi = 0.9$. The boxplots are obtained over 100 replications.

Spherical Harmonics as Control Variates

4

The Sliced-Wasserstein (SW) distance between probability measures is defined as the average of the Wasserstein distances resulting for the associated one-dimensional projections. As a consequence, the SW distance can be written as an integral with respect to the uniform measure on the sphere and the Monte Carlo framework can be employed for calculating the SW distance. Spherical harmonics are polynomials on the sphere that form an orthonormal basis of the set of square-integrable functions on the sphere. Putting these two facts together, a new Monte Carlo method, hereby referred to as Spherical Harmonics Control Variates (SHCV), is proposed for approximating the SW distance using spherical harmonics as control variates. The resulting approach is shown to have good theoretical properties, e.g., a no-error property for Gaussian measures under a certain form of linear dependency between the variables. Moreover, an improved rate of convergence, compared to Monte Carlo, is established for general measures. The convergence analysis relies on the Lipschitz property associated to the SW integrand. Several numerical experiments demonstrate the superior performance of SHCV against state-of-the-art methods for SW distance computation. The content of this chapter is based on the paper

R. Leluc, A. Dieuleveut, F. Portier, J. Segers and A. Zhuman. (2024). Sliced-Wasserstein Estimation with Spherical Harmonics as Control Variates. *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of PMLR, pages 27191-27214.

4.1 Introduction

The Sliced-Wasserstein (SW) distance between two probability measures, introduced in Rabin et al. (2012), shares similar topological properties with the standard Wasserstein distance (Bonnotte, 2013; Bayraktar and Guo, 2021; Nadjahi et al., 2020b) while having better properties in terms of computational complexity, especially for measures defined on spaces with relatively large dimension (Nadjahi et al., 2020b). As such, it has emerged as a power-

ful framework for analyzing and solving a wide range of machine learning problems including generative modeling (Deshpande et al., 2018, 2019; Litkus et al., 2019), autoencoders (Kolouri et al., 2018), Bayesian computation (Nadjahi et al., 2020a), and image processing (Bonneel et al., 2015).

Informally, for $p \geq 1$ the SW_p distance between probability distributions μ and ν on $\mathbb{R}^d$ is defined as the integral value, over θ uniformly distributed on the unit sphere $\mathbb{S}^{d-1}$, of the Wasserstein distance W_p between the projections of distributions μ and ν on the direction of θ . It takes advantage of the fact that (a) the Wasserstein distance between one-dimensional measures has a simple analytical form, being the $L^p([0, 1])$ -distance between their quantile functions (approximated using empirical quantiles), and (b) the integral is approximated by Monte Carlo sampling (uniformly on the unit sphere). As a consequence, two types of error affect the accuracy of most algorithms estimating the SW distance: first, the statistical error, due to the approximation of the distributions μ and ν by their empirical counterparts, which scales as $m^{-1/2}$, with m the size of the samples observed from μ and ν (Nietert et al., 2022); second, the Monte Carlo integration error, which scales as $n^{-1/2}$, with n the number of projections, as demonstrated in Nadjahi et al. (2020b) and Nietert et al. (2022). Importantly, these results establish that the sample complexity to approximate the SW distance does not depend on the dimension. As pointed out in Nadjahi et al. (2020b), even though the SW sample complexity is reduced compared to Wasserstein, it requires an additional Monte Carlo step which might offset the previous benefits if the associated variance is large.

In this chapter, we propose and study a new approach to improve SW distance computation by improving the Monte Carlo estimation (step (b) above). The approach follows from a well-known variance reduction principle introducing *control variates* (functions having known integrals) in the Monte Carlo estimate. Following recent papers Oates et al. (2017); Portier and Segers (2019); Leluc et al. (2021); South et al. (2023), the use of many control variates can significantly reduce the variance of the Monte Carlo method when the integrand f is well-approximated in some L^2 -space by the control variates. In Portier and Segers (2019) for instance, the obtained convergence rate is $n^{-1/2}\sigma_{cv}$, where σ_{cv}^2 is the variance of the residual function resulting from approximating f in the control variate space. By allowing the number of control variates to grow with n , the residual standard deviation σ_{cv} can vanish asymptotically, leading to an improvement of the $n^{-1/2}$ rate of the Monte Carlo estimate.

As the integral of interest in the SW distance is over the sphere, we propose to use as control variates the well-known *spherical harmonics*, which are homogeneous orthogonal polynomials defined on the sphere. Our approach,

referred to as Sliced Wasserstein estimates with Spherical Harmonics as Control Variates (SHCV), leverages the property of spherical harmonics to be an orthonormal basis of $L_2(\mathbb{S}^{d-1})$ to build a precise and flexible approximation of the integrand leading to a more accurate Monte Carlo estimate.

Related methods. Several methods have been introduced for computing the SW distance and improve upon the standard Monte Carlo estimate. Nguyen and Ho (2024) use a *single* quadratic control variate to reduce the variance. This approach significantly differs from SHCV, which relies on a potentially large number of control variates. Further, for functions defined on $\mathbb{R}^d$, Leluc et al. (2024b) use the *leave-one-out* nearest neighbor regression estimate as a control variate. As soon as the integrand is Lipschitz, the rate of convergence is $n^{-1/2-1/d}$, which improves upon the $n^{-1/2}$ rate of the Monte Carlo estimate. Even though this technique has not been studied for other spaces than $\mathbb{R}^d$ (such as the sphere), good practical performance has been reported in Leluc et al. (2024b) for approximating the SW distance.

In an attempt to improve the Monte Carlo estimate, Quasi-Monte Carlo (QMC) and Randomized Quasi Monte Carlo (RQMC) methods have been proposed to approximate the SW distance (Nguyen et al., 2024) based on deterministic sequences that are well spread over the integration domain. Finally, we mention the orthogonal approach of Nadjahi et al. (2021) which is based on Gaussian approximation as the dimension d grows, thus targets large dimension, and only applies for $p = 2$.

Contributions. The main contributions of this chapter are summarized as follows:

- We propose a novel variance-reduced SW estimate based on control variate with spherical harmonics.
- We derive some theoretical properties as well as the convergence rate in probability on the SHCV integration error, providing rigorous underpinnings for the proposed methodology. Our analysis reveals that SHCV can achieve faster convergence rates than traditional unbiased estimators of the SW distance.
- In numerical experiments, our approach outperforms not only existing Monte Carlo approaches based on different control variates but also some recently introduced quasi Monte Carlo (QMC) methods (see related methods described above), without prohibitive additional cost.

Outline. Section 4.2 introduces the mathematical background of (Sliced-)Wasserstein distances while Section 4.3 presents MC integration with control variates. In Section 4.4, we introduce our estimator, based on spherical harmonics. It is empirically evaluated in Section 4.5. Section 4.6 provides a

theoretical analysis and Section 4.7 concludes the chapter. Section 4.8 contain proofs, algorithms and additional numerical results.

Notation. For a Borel set $X \subseteq \mathbb{R}^d$, $\mathcal{P}(X)$ is the set of probability measures with support contained in X . Let $p \in [1, \infty)$ and $\mathcal{P}_p(X) = \{\mu \in \mathcal{P}(X) : \int_X \|x\|^p d\mu(x) < +\infty\}$ be the set of probability measures on X with finite moment of order p . The unit sphere on $\mathbb{R}^d$ is $\mathbb{S}^{d-1} = \{\theta \in \mathbb{R}^d : \|\theta\| = 1\}$ with surface area $|\mathbb{S}^{d-1}| = (2\pi^{d/2})/\Gamma(d/2)$, where Γ is the Euler gamma function. $\langle \cdot, \cdot \rangle$ is the Euclidean inner product and $\mathcal{N}(m, \Sigma)$ is the d -Gaussian distribution with mean $m \in \mathbb{R}^d$ and positive definite $d \times d$ covariance matrix Σ . The Laplace operator is $\Delta = \partial_1^2 + \dots + \partial_d^2$ where ∂_i is the i -th partial derivative.

4.2 Optimal transport distances: Wasserstein and sliced-Wasserstein

4.2.1 Wasserstein distance

The Wasserstein distance $W_p(\mu, \nu)$ of order $p \in [1, \infty)$ between probability measures $\mu, \nu \in \mathcal{P}_p(\mathbb{R}^d)$ is defined by

$$W_p^p(\mu, \nu) = \inf_{\pi \in \Pi(\mu, \nu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\|^p d\pi(x, y),$$

where $\Pi(\mu, \nu) \subset \mathcal{P}(\mathbb{R}^d \times \mathbb{R}^d)$ denotes the set of couplings for (μ, ν) , i.e., probability measures whose marginals with respect to the first and second variables are μ and ν , respectively. While the Wasserstein distance enjoys attractive theoretical properties (Villani, 2009, Chapter 6), it suffers from a high computational cost. When computing $W_p(\mu_m, \nu_m)$ for discrete distributions μ_m and ν_m supported on m points, the worst-case computational complexity scales as $O(m^3 \log m)$ (Peyré et al., 2019). However, for *univariate* distributions $\mu, \nu \in \mathcal{P}_p(\mathbb{R})$, the Wasserstein distance can be expressed through the quantile functions F^{-1} (inverse c.d.f.) as (see Peyré et al., 2019, Section 2)

$$W_p^p(\mu, \nu) = \int_0^1 |F_\mu^{-1}(t) - F_\nu^{-1}(t)|^p dt.$$

In particular, for discrete measures $\mu_m = m^{-1} \sum_{i=1}^m \delta_{x_i}$ and $\nu_m = m^{-1} \sum_{i=1}^m \delta_{y_i}$ with $x_i, y_i \in \mathbb{R}$, the Wasserstein distance can be computed by sorting the atoms, yielding

$$W_p^p(\mu_m, \nu_m) = \frac{1}{m} \sum_{i=1}^m |x_{(i)} - y_{(i)}|^p, \quad (4.1)$$

where $(x_{(i)})_{i=1}^m$ and $(y_{(i)})_{i=1}^m$ are the order statistics of $(x_i)_{i=1}^m$ and $(y_i)_{i=1}^m$ respectively. The number of operations induced by the sorting step is $O(m \log m)$.

4.2.2 Sliced-Wasserstein distance

The Sliced-Wasserstein distance (Rabin et al., 2012; Bonneel et al., 2015; Kolouri et al., 2019) alleviates the high computational cost of Wasserstein distance, by *slicing* distributions and by taking advantage of the fast computation of the Wasserstein distance between *univariate* distributions (see Equation (4.1)).

Definition 4.1 (Sliced-Wasserstein distances). For $\theta \in \mathbb{S}^{d-1}$, let $\theta^\star : \mathbb{R}^d \rightarrow \mathbb{R}$ denote the linear functional $\theta^\star(x) = \langle \theta, x \rangle$ for $x \in \mathbb{R}^d$. Let $P \in \mathcal{P}(\mathbb{S}^{d-1})$ be a probability distribution on the unit sphere. The SW distance of order $p \in [1, \infty)$ based on P is defined for $\mu, \nu \in \mathcal{P}_p(\mathbb{R}^d)$ as

$$\text{SW}_p^p(\mu, \nu, P) = \int_{\mathbb{S}^{d-1}} W_p^p(\theta_\#^\star \mu, \theta_\#^\star \nu) dP(\theta), \quad (4.2)$$

where for any measure $\xi \in \mathcal{P}(\mathbb{R}^d)$, $\theta_\#^\star \xi$ is the push-forward measure by function $\theta^\star$, i.e. the distribution of the projection $\theta^\star(X)$ of a random vector $X \in \mathbb{R}^d$ with distribution ξ .

We introduce the integrand $f_{\mu, \nu}^{(p)} : \mathbb{S}^{d-1} \rightarrow \mathbb{R}$ defined by

$$\forall \theta \in \mathbb{S}^{d-1}, \quad f_{\mu, \nu}^{(p)}(\theta) = W_p^p(\theta_\#^\star \mu, \theta_\#^\star \nu). \quad (4.3)$$

In practice, the SW distance of Equation (4.2) is approximated using a standard Monte Carlo estimate by (i) sampling a number $n \geq 1$ of independent random directions $\theta_1, \dots, \theta_n$ from P on $\mathbb{S}^{d-1}$ and (ii) by averaging the values $(f_{\mu, \nu}^{(p)}(\theta_i))_i$. This is described in Algorithm 4.1. In the chapter, we will often work with the uniform distribution on the sphere, denoted by U or $\mathcal{U}(\mathbb{S}^{d-1})$.

Algorithm 4.1 Sliced Wasserstein Monte Carlo

Require: $\mu, \nu \in \mathcal{P}_p(\mathbb{R}^d)$, number of random projections n , probability $P \in \mathcal{P}(\mathbb{S}^{d-1})$

- 1: Sample random projections $\theta_1, \dots, \theta_n \sim P$
 - 2: Compute $f_n = (f_{\mu, \nu}^{(p)}(\theta_i))_{i=1}^n$
 - 3: Return average $\text{MC}_n = (\mathbf{1}_n/n)^\top f_n$
-

Computing the SW distance between discrete measures $\mu_m = m^{-1} \sum_{i=1}^m \delta_{x_i}$ and $\nu_m = m^{-1} \sum_{i=1}^m \delta_{y_i}$ with atoms $x_i, y_i \in \mathbb{R}^d$ amounts to projecting $(x_i)_{i=1}^m$ and $(y_i)_{i=1}^m$ along the n random directions $\theta_1, \dots, \theta_n$ followed by computing the one-dimensional Wasserstein distances using Equation (4.1). This scheme requires $O(n(dm + m \log m))$ operations which is, in general, faster than computing $W_p^p(\mu_m, \nu_m)$, especially for large m .

Remark 4.1 (Gaussian case when $p = 2$). The Wasserstein distance between two Gaussians $\mu = \mathcal{N}(a, \mathbf{A})$ and $\nu = \mathcal{N}(b, \mathbf{B})$ is known in closed form (Dowson and Landau, 1982) as $W_2^2(\mu, \nu) = \|a - b\|_2^2 + \mathfrak{B}^2(\mathbf{A}, \mathbf{B})$ where $\mathfrak{B}$ is the Bures distance (Bhatia et al., 2019) on positive definite matrices, $\mathfrak{B}^2(\mathbf{A}, \mathbf{B}) = \text{Tr}(\mathbf{A}) + \text{Tr}(\mathbf{B}) - 2 \text{Tr}[(\mathbf{A}^{1/2} \mathbf{B} \mathbf{A}^{1/2})^{1/2}]$. The Sliced-Wasserstein distance between μ and ν relies on the projected distributions $\theta_\#^\star \mu \sim \mathcal{N}(\theta^\top a, \theta^\top \mathbf{A} \theta)$ and $\theta_\#^\star \nu \sim \mathcal{N}(\theta^\top b, \theta^\top \mathbf{B} \theta)$ via the integration over $\theta \in \mathbb{S}^{d-1}$ of $W_2^2(\theta_\#^\star \mu, \theta_\#^\star \nu) = |\theta^\top(a - b)|^2 + (\sqrt{\theta^\top \mathbf{A} \theta} - \sqrt{\theta^\top \mathbf{B} \theta})^2$.

4.3 Monte Carlo with control variates

This section presents the mathematical framework of Monte Carlo integration with control variates.

Monte Carlo integration. Consider a square-integrable, real-valued function $f \in L_2(\mathbb{P})$ on a probability space $(\Theta, \mathcal{F}, \mathbb{P})$ of which we would like to compute the integral

$$I(f) = \int_{\Theta} f(\theta) d\mathbb{P}(\theta). \quad (4.4)$$

When $I(f)$ does not admit a closed form or when only calls from f are available, one may rely on Monte Carlo estimates. Let $\theta_1, \dots, \theta_n \sim \mathbb{P}$ be an independent and identically distributed (i.i.d.) random sample from $\mathbb{P}$. The naive Monte Carlo estimate $I_n(f)$ of $I(f)$ is given by the empirical mean

$$I_n(f) = \frac{1}{n} \sum_{i=1}^n f(\theta_i).$$

The Monte Carlo estimate is unbiased and has variance equal to $\sigma^2(f)/n$ where $\sigma^2(f) = \mathbb{I}[f - I(f)]^2$. Increasing the sample size n reduces the variance but at the cost of an increased computation time. Instead, it is possible to find another estimate with smaller variance.

Control variates and OLSMC. A classical way to reduce the variance of the Monte Carlo estimate consists in incorporating knowledge about some specially chosen functions. Control variates are functions $\varphi_1, \dots, \varphi_s : \Theta \rightarrow \mathbb{R}$ with known expectations; without loss of generality, we assume that $I(\varphi_j) = 0$ for $j = 1, \dots, s$. Let $\varphi = (\varphi_1, \dots, \varphi_s)^\top$ denote the $\mathbb{R}^s$ -valued function with s control variates as elements. Since all control variates are centered, adding any linear combination of the form $\beta^\top \varphi$ with $\beta \in \mathbb{R}^s$ gives $I(f - \beta^\top \varphi) = I(f)$. Thus, given an i.i.d. sample $\theta_1, \dots, \theta_n$ from $\mathbb{P}$, any coefficient vector $\beta \in \mathbb{R}^s$

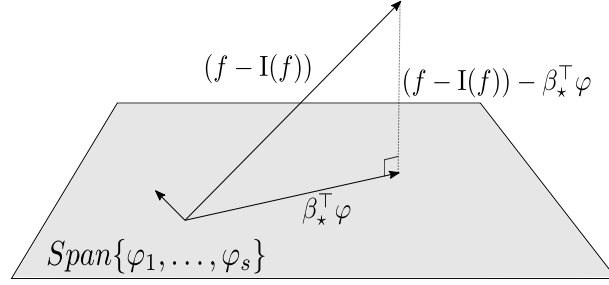


Figure 4.1: Visualization of the L_2 -projection of the integrand f onto the linear space $\text{Span}\{\varphi_1, \dots, \varphi_s\}$ of control variates.

yields an unbiased estimate of $I(f)$, given by

$$I_n(f, \beta) = \frac{1}{n} \sum_{i=1}^n (f(\theta_i) - \beta^\top \varphi(\theta_i)), \quad (4.5)$$

with variance equal to $\sigma^2(f - \beta^\top \varphi)/n$. The optimal coefficient $\beta_\star$ is defined as the one minimizing the variance,

$$\beta_\star \in \arg \min_{\beta \in \mathbb{R}^s} \mathbb{I}[\{f - I(f) - \beta^\top \varphi\}^2].$$

By the Hilbert projection theorem, $\beta_\star^\top \varphi$ is the $L_2(\mathbb{P})$ -projection of $f - I(f)$ on the linear space of control variates $\text{Span}\{\varphi_1, \dots, \varphi_s\}$, as illustrated in Figure 4.1.

Different approaches have been proposed to estimate $\beta_\star$, each of which, when used in Equation (4.5), produces an enhanced Monte Carlo estimate with reduced variability (Glynn and Szechtman, 2002). Following Owen (2013, Section 8.9), or more recently Portier and Segers (2019) and Leluc et al. (2021), we frame the problem as an ordinary least-squares minimization problem. The integral $I(f)$ appears as the intercept of the linear regression model with explanatory variables $\varphi_1, \dots, \varphi_s$ and target response f , i.e.,

$$(I(f), \beta_\star(f)) \in \arg \min_{(\alpha, \beta) \in \mathbb{R} \times \mathbb{R}^s} \mathbb{I}[(f - \alpha - \beta^\top \varphi)^2].$$

In practice, these quantities are estimated using the directions $\theta_1, \dots, \theta_n$ sampled from $\mathbb{P}$. The Ordinary Least Squares Monte Carlo (OLSMC) estimate $I_n^{\text{ols}}(f)$ is¹

$$(I_n^{\text{ols}}(f), \beta_n(f)) \in \arg \min_{(\alpha, \beta) \in \mathbb{R} \times \mathbb{R}^s} \|f_n - \alpha \mathbf{1}_n - \Phi \beta\|_2^2, \quad (4.6)$$

¹It is well-defined if and only if $\mathbf{1}_n \notin \text{Span}(\Phi)$. If ill-defined, one can reduce the number of control variates such that $s < n$.

where $f_n = (f(\theta_1), \dots, f(\theta_n))^\top \in \mathbb{R}^n$ is the vector of evaluations of f , $\mathbf{1}_n = (1, \dots, 1)^\top \in \mathbb{R}^n$ and $\Phi \in \mathbb{R}^{n \times s}$ is the random matrix of control variates $\Phi = (\varphi(\theta_i)^\top)_{i=1}^n$.

Remark 4.2 (Complexity). *The Monte Carlo estimate requires $O(n\omega_f)$ computations where ω_f is the cost of evaluating the integrand f . The OLSMC estimate has a complexity of $O(n\omega_f + \omega(\Phi))$ where $\omega(\Phi)$ is the complexity due to the additional step of fitting the optimal control variates. Typically, $\omega(\Phi) = O(ns^2 + s^3)$ where $O(ns^2)$ and $O(s^3)$ operations are needed to compute and invert $\Phi^\top \Phi$, though it can be reduced by using an approximate solution via optimization algorithms such as (stochastic) gradient descent algorithms (Zhang, 2004). When f is heavy to compute—which is the case for $f_{\mu_m, \nu_m}^{(p)}$ with discrete measures supported on a large number m of atoms—the extra term $\omega(\Phi)$ becomes negligible compared to $O(n\omega_f)$.*

Remark 4.3 (Regularization). *As the number s of control variates may increase quickly with the dimension, Leluc et al. (2021) rely on the LASSO procedure (Tibshirani, 1996) in place of Equation (4.6) to achieve estimation and variable selection at the same time in the regime $n \ll s$.*

4.4 Sliced-Wasserstein estimate with spherical harmonics as control variates

4.4.1 Spherical harmonics

Originating from the solutions to the Laplace equation on the unit sphere, spherical harmonics (Atkinson and Han, 2012; Dai, 2013) constitute a mathematical framework with applications in computer graphics (Ramamoorthi and Hanrahan, 2001; Basri and Jacobs, 2003; Green, 2003) and machine learning (Cohen et al., 2018; Dutordoir et al., 2020), particularly in tasks involving spherical data representation.

Definition 4.2 (Polynomial spaces). *Let $\mathcal{P}_\ell^d$ be the space of homogeneous polynomials of degree $\ell \geq 0$ on $\mathbb{R}^d$, i.e., $\mathcal{P}_\ell^d = \text{Span}\{x_1^{a_1} \cdots x_d^{a_d} \mid a_k \in \mathbb{N}, \sum_{k=1}^d a_k = \ell\}$. Let $\mathcal{H}_\ell^d$ be the space of real harmonic polynomials, homogeneous of degree ℓ on $\mathbb{R}^d$, i.e., $\mathcal{H}_\ell^d = \{Q \in \mathcal{P}_\ell^d \mid \Delta Q = 0\}$.*

Definition 4.3 (Spherical harmonics). *Spherical harmonics of degree (or level) $\ell \geq 0$ are defined as the restriction of elements in $\mathcal{H}_\ell^d$ to the unit sphere $\mathbb{S}^{d-1}$, i.e., restrictions to the sphere of harmonic homogeneous polynomials with d variables of degree ℓ .*

Let N_ℓ^d denote the number of linearly independent spherical harmonics of degree $\ell \geq 0$ in dimension d , that is $N_\ell^d = \dim \mathcal{H}_\ell^d = \dim \mathcal{P}_\ell^d - \dim \mathcal{P}_{\ell-2}^d$

(Dai, 2013, Corollary 1.1.4) where it is agreed that $\dim \mathcal{P}_{\ell-2}^d = 0$ for $\ell = 0, 1$. We have

$$N_\ell^d = \frac{(2\ell + d - 2)(\ell + d - 3)!}{\ell!(d - 2)!}.$$

Note that, when restricted to the sphere, all polynomials are linear combinations of spherical harmonics. Moreover, spherical harmonics of different degrees are orthogonal with respect to $\langle f, g \rangle = \int_{\mathbb{S}^{d-1}} f(\theta)g(\theta) dU(\theta)$, where U is the uniform distribution on $\mathbb{S}^{d-1}$ (Dai, 2013, Theorem 1.1.2). In practice, orthonormal bases can be constructed by applying the Gram-Schmidt process and an explicit basis of spherical harmonics can be written in terms of the Gegenbauer polynomials in spherical coordinates (Dai, 2013, Theorem 1.5.1). In the following, let $\varphi_{\ell,k}$ with $\ell \geq 0$ and $k = 1, \dots, N_\ell^d$ denote a set of spherical harmonics forming an orthonormal basis of $\mathcal{H}_\ell^d$.

Proposition 4.1 (Hilbert basis). *The spherical harmonics $\{\varphi_{\ell,k}, 0 \leq \ell, 1 \leq k \leq N_\ell^d\}$ form an orthonormal eigenbasis of the Hilbert space $L_2(\mathbb{S}^{d-1}, U)$, so that for every $f \in L_2(\mathbb{S}^{d-1})$ we have*

$$f = \sum_{\ell=0}^{\infty} \sum_{k=1}^{N_\ell^d} \hat{f}_{\ell,k} \varphi_{\ell,k} \quad \text{where} \quad \hat{f}_{\ell,k} = \int f \varphi_{\ell,k} dU.$$

Spherical harmonics thus constitute an eigenbasis of centered functions, and can therefore be used as control variates with respect to the uniform distribution on the unit sphere. For $\ell \geq 1$ and $k = 1, \dots, N_\ell^d$, we have

$$I(\varphi_{\ell,k}) = \int_{\mathbb{S}^{d-1}} \varphi_{\ell,k}(\theta) dU(\theta) = 0. \quad (4.7)$$

Orthonormality ensures that the empirical covariance matrix is well-conditioned when solving the OLS problem (4.6).

Remark 4.4 (Examples). *In most applications, spherical harmonics in two and three variables are used. For $d = 2$, we have $\dim \mathcal{H}_\ell^2 = 2$ for all $\ell \geq 1$ and an orthogonal basis is given by the real and imaginary parts of $(x + iy)^\ell$. In polar coordinates $(x, y) = (r \cos(t), r \sin(t)) \in \mathbb{R}^2$, this basis is $\varphi_{\ell,1}(x, y) = r^\ell \cos(\ell t)$ and $\varphi_{\ell,2}(x, y) = r^\ell \sin(\ell t)$. By restriction on the circle $\mathbb{S}^1$, spherical harmonics yield the classical Fourier expansions in cosine and sine functions and are commonly referred to as circular harmonics. For $d = 3$, we have $\dim \mathcal{H}_\ell^3 = 2\ell + 1$ and a classical orthogonal basis in spherical coordinates is written in terms of the associated Legendre polynomials (see Section 4.8.4).*

4.4.2 Spherical harmonics control variates estimate

With the idea of performing variance reduction when computing SW distances, we incorporate spherical harmonics as natural control variates on $\mathbb{S}^{d-1}$ into the standard Monte Carlo estimate. We propose a novel procedure called *Spherical Harmonics Control Variates* (SHCV) estimate using the ordinary least squares formulation of Equation (4.6).

For $p \geq 1$, consider $\mu, \nu \in \mathcal{P}_p(\mathbb{R}^d)$ and recall the integrand $f_{\mu,\nu}^{(p)} : \theta \mapsto W_p^p(\theta_\#^* \mu, \theta_\#^* \nu)$ so that $\text{SW}_p^p(\mu, \nu) = \mathbb{I}(f_{\mu,\nu}^{(p)})$. Interestingly, this integrand is even so that for odd functions φ , the inner product vanishes, $\langle f_{\mu,\nu}^{(p)}, \varphi \rangle = 0$. The spherical harmonics expansion of $f_{\mu,\nu}^{(p)}$ (see Proposition 4.1) is only made of spherical harmonics with even degree. Thus, when finding the optimal control variates among spherical harmonics, we can omit the ones of odd degree as these are uncorrelated with our integrand.

Lemma 4.1 (Number of control variates). *In dimension d , the dimension of the linear space generated by the spherical harmonics of even degree up to maximal degree $2L$ is*

$$s_{L,d} = \sum_{\ell=1}^L N_{2\ell}^d = \binom{2L+d-1}{d-1} - 1.$$

For $d = 2$, the formula reduces to $s_{L,2} = 2L$ which is the number of circular harmonics of degree $2, 4, \dots, 2L$; For $d = 3$, the formula becomes $s_{L,3} = (2L + 3)L$, which is indeed the sum of $(2\ell + 1)$ for $\ell = 2, 4, \dots, 2L$ (Remark 4.4).

Definition 4.4 (SHCV). *For $\mu, \nu \in \mathcal{P}_p(\mathbb{R}^d)$, the SHCV estimate of maximum degree $2L$ based on projections $\theta_1, \dots, \theta_n \in \mathbb{S}^{d-1}$ and control variates $\varphi = (\varphi_j)_{j=1}^{s_{L,d}}$ is defined as the intercept of the OLS problem (4.6) with the integrand $f_{\mu,\nu}^{(p)}$ as response and all spherical harmonics of even degree from 2 up to $2L$ as $n \times s_{L,d}$ covariate matrix $\Phi = (\varphi(\theta_i)^\top)_{i=1}^n$, i.e.,*

$$\text{SHCV}_{n,L}^p(\mu, \nu) = \mathbb{I}_n^{\text{ols}}(f_{\mu,\nu}^{(p)}). \quad (4.8)$$

The procedure is described in Algorithm 4.2. In the algorithm, the number of spherical harmonics is specified in advance. A selection based on a lasso procedure with cross-validation for the penalization parameter would be possible too, but is not pursued here for the sake of computing efficiency.

Remark 4.5 (Multiple integrals). *Computational benefits occur when there are multiple integrands, since the SHCV estimate can be represented as a linear rule $w^\top f_n$ (Portier and Segers, 2019; Leluc et al., 2021), where the weight vector $w \in \mathbb{R}^n$ does not depend on the integrands. If K integrals need to be evaluated, the*

Algorithm 4.2 Spherical Harmonics Control Variate Estimate

Require: $\mu, \nu \in \mathcal{P}_p(\mathbb{R}^d)$, number of random projections n , $P \in \mathcal{P}(\mathbb{S}^{d-1})$, spherical harmonics $\varphi = (\varphi_j)_{j=1}^s$

- 1: Sample random projections $\theta_1, \dots, \theta_n \sim P$
- 2: Compute $f_n = (f_{\mu, \nu}^{(p)}(\theta_i))_{i=1}^n$, $\Phi = (\varphi(\theta_i)^\top)_{i=1}^n$
- 3: Solve $(\mathbf{I}_n^{\text{ols}}, \beta_n) \in \arg \min_{\alpha, \beta} \|f_n - \alpha \mathbf{1}_n - \Phi \beta\|_2^2$
- 4: Return **SHCV** $_n = \mathbf{I}_n^{\text{ols}}$

computation time of SHCV becomes $O(Kn\omega_f + \omega(\Phi))$ compared to $O(Kn\omega_f)$ for standard Monte Carlo, and the additional cost $\omega(\Phi)$ becomes negligible for large K . This is particularly relevant for SW-based kernels involving many pairwise computations $\text{SW}_2^2(\mu_i, \nu_j)$; see Algorithms 4.9 and 4.10 in Section 4.8.2.

Remark 4.6 (Quadratic control variates). Any symmetric $d \times d$ matrix M yields a quadratic control variate $\varphi_M(\theta) = \theta^\top M \theta - \text{Tr}(M)/d$ as $\mathbb{I}(\varphi_M) = 0$. This is actually the approach used in Nguyen and Ho (2024) where the authors propose two different choices for M leading to two different estimators (see Section 4.8.1). Note that, whenever $2L \geq 2$, these quadratic functions are included in our spherical harmonics space, making our method more general with potentially greater variance reduction (see Section 4.6).

4.5 Numerical experiments

To assess the finite-sample performance of the SHCV estimate, we first present some synthetic data examples involving the computation of $\text{SW}_2^2(\mu, \nu)$ between multivariate Gaussian distributions. We consider different dimensions $d \in \{3; 5; 6; 10; 20\}$ and a varying number of random projections $n \in [10^2; 10^4]$. Then we focus on $\text{SW}_2^2(\mu_m, \nu_m)$ between 3D point clouds from the ShapeNetCore dataset of Chang et al. (2015). In both cases, the ground truth of the Sliced-Wasserstein distance is estimated with a very large number $n = 10^8$ of projections. The different results report the mean squared error (MSE) $\mathbb{E}[|\text{SW}_{2,n}^2(\mu, \nu) - \text{SW}_2^2(\mu, \nu)|^2]$ for the different estimates $\text{SW}_{2,n}^2(\mu, \nu)$ where the expectation is computed as an average over 100 independent runs. Finally, in the spirit of Kolouri et al. (2016) and Meunier et al. (2022), we implement a Kernel Support Vector Machine (SVM) classifier for image classification. For ease of reproducibility, numerical details are in the Appendix and the code is available here². In particular, we rely on the efficient implementation of Dutordoir et al. (2020) to build the spherical harmonics. The degrees of the spherical harmonics are reported in Table 4.9 in Section 4.8.4.

²<https://github.com/RemiLELUC/SHCV>

Methods in competition. In the experiments, the SW distance is computed through different estimates. Along with the standard Monte Carlo estimate and the proposed SHCV estimate, we consider three other baselines based on control variates: the two estimates of Nguyen and Ho (2024) based on lower and upper bounds of a Gaussian approximation and the *control neighbors* estimate of Leluc et al. (2024b) based on nearest neighbors estimates acting as control variates. For the sake of completeness, we also include a comparison with (Randomized) Quasi Monte Carlo procedures as in Nguyen et al. (2024) where deterministic QMC sets from $[0, 1]^d$ are mapped to the unit sphere and randomly rotated. The methods in competition are: the standard Monte Carlo estimate (MC), the lower-CV (CV_{low}) and upper-CV (CV_{up}) estimates, the control neighbors estimate (CVNN), the proposed estimate with spherical harmonics as control variates (SHCV), the Quasi Monte Carlo (QMC) and Random Quasi Monte Carlo estimates (RQMC).

4.5.1 Synthetic data: multivariate Gaussians

We compute $SW_2^2(\mu_m, \nu_m)$ between empirical distributions $\mu_m = m^{-1} \sum_{i=1}^m \delta_{x_i}$ and $\nu_m = m^{-1} \sum_{j=1}^m \delta_{y_j}$ where $x_1, \dots, x_m \sim \mu = \mathcal{N}(a, \mathbf{A})$, $y_1, \dots, y_m \sim \nu = \mathcal{N}(b, \mathbf{B})$ with $m = 1000$ samples. The parameters of the Gaussians are $a, b \sim \mathcal{N}(1_d, I_d)$ and covariance matrices $\mathbf{A} = \Sigma_a \Sigma_a^\top$ and $\mathbf{B} = \Sigma_b \Sigma_b^\top$ where all entries of Σ_a, Σ_b are drawn according to $\mathcal{N}(0, 1)$. In Section 4.8.2, we also consider the *exact case* where we directly evaluate the smooth integrand $f_{\mu, \nu}^{(2)}$ given in Remark 4.1. In that case, as the original lower-CV (CV_{low}) and upper-CV (CV_{up}) estimates are written to take discrete measures as inputs, we adapt them with the true means and covariance matrices.

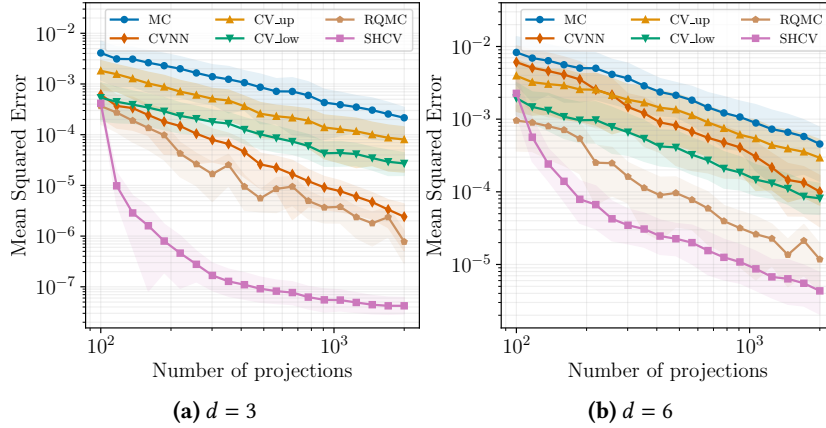


Figure 4.2: MSE curves for sampled Gaussian distributions in dimension $d \in \{3; 6\}$.

Figure 4.2 reports the MSE of the different methods in dimension $d \in$

$\{3; 6\}$ with respect to the number of projections, while Figure 4.5 in Section 4.8.2 does so with respect to computing time. Table 4.1 reports the MSE and computing times (in ms) in higher dimension $d \in \{5; 10; 20\}$. Figure 4.4 and Table 4.3 in Section 4.8.2 deal with the *exact case* accordingly. In each case, the SHCV estimate gives the best MSE performance with respect to both the number of projections and computing time, beating other baselines by a factor up to 100.

Method	$d = 5$		$d = 10$		$d = 20$	
	MSE	Time	MSE	Time	MSE	Time
MC	1.45e-3	81.1 $\pm$ 3.5	9.45e-4	80.7 $\pm$ 4.4	1.47e-3	81.1 $\pm$ 1.8
CV _{low}	2.67e-4	79.7 $\pm$ 1.1	3.45e-4	80.1 $\pm$ 1.4	3.82e-4	80.0 $\pm$ 1.0
CV _{up}	8.44e-4	83.0 $\pm$ 1.2	7.51e-4	83.0 $\pm$ 1.7	1.09e-3	83.1 $\pm$ 1.5
CVNN	4.29e-4	110 $\pm$ 2.2	1.12e-3	122 $\pm$ 1.6	2.14e-3	127 $\pm$ 1.4
QMC	2.91e-4	100 $\pm$ 1.2	2.37e-4	113 $\pm$ 1.4	6.60e-4	129 $\pm$ 1.4
RQMC	5.80e-5	96.3 $\pm$ 2.2	2.75e-4	113 $\pm$ 1.2	1.17e-3	130 $\pm$ 1.0
SHCV	2.68e-6	89.0 $\pm$ 6.3	1.93e-4	89.0 $\pm$ 4.5	2.95e-4	88.1 $\pm$ 2.8

Table 4.1: MSE and computing time (ms) for Gaussian distributions in dimension $d \in \{5; 10; 20\}$ based on $n = 500$ projections.

4.5.2 Empirical distributions of 3D point clouds

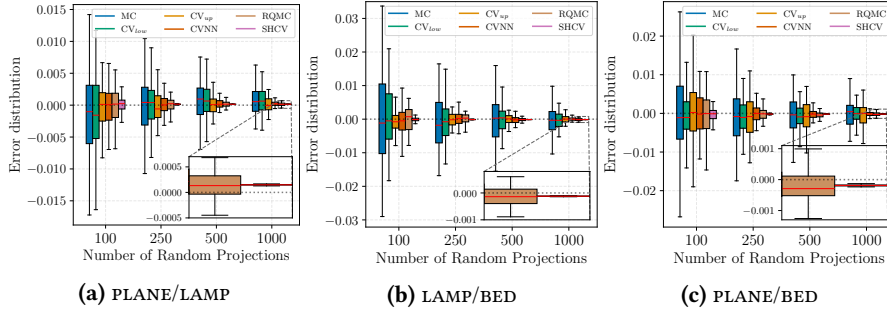


Figure 4.3: Boxplots of the error distribution $\widehat{SW}_n(\mu_m, v_m) - SW(\mu_m, v_m)$ for different SW estimates based on n random projections with $n \in \{100; 250; 500; 1000\}$ obtained over 100 independent runs.

Similarly to Nguyen and Ho (2024), we use three point clouds taken at random from the dataset ShapeNetCore (Chang et al., 2015) corresponding to the objects plane, lamp, and bed, each composed of $m = 2048$ points in $\mathbb{R}^3$. Each point cloud $(x_1, \dots, x_m)$ is encoded as a discrete measure $\mu_m = m^{-1} \sum_{i=1}^m \delta_{x_i}$ with equal masses over the points in the cloud and the goal is to compute

$SW(\mu_m, \nu_m)$ of order $p = 2$ between empirical distributions. Figure 4.3 reports the boxplots of the error distribution $\widehat{SW}_n(\mu_m, \nu_m) - SW(\mu_m, \nu_m)$ for the different SW estimates based on n random projections with $n \in \{100; 250; 500; 1000\}$ obtained over 100 independent runs. Once again, our SHCV estimate provides huge gains in terms of variance reduction and outperforms other baselines by a factor 100; see Table 4.6 in Section 4.8.2.

4.5.3 Sliced-Wasserstein and Kernel SVM

SW distances are useful in the context of distribution regression (Szabó et al., 2016) where the goal is to learn a real-valued function defined on the space of probability distributions from a sequence of observations. Kolouri et al. (2016) first introduced a SW-based kernel for absolutely continuous distributions, which was later on extended to empirical distributions by Meunier et al. (2022). For any $\gamma > 0$, both $e^{-\gamma SW_2^2(\cdot, \cdot)}$ and $e^{-\gamma SW_1(\cdot, \cdot)}$ are valid kernels on probability measures. Using the SW kernel of Kolouri et al. (2016) defined by $k(\mu_i, \mu_j) = \exp(-\gamma SW_2^2(\mu_i, \mu_j))$ with $\gamma > 0$, we consider the image classification task on the digits dataset of Alpaydin and Kaynak (1998). This dataset contains $N = 1797$ images of size 8×8 and the goal is to predict the label $z \in \{0, 1, \dots, 9\}$ of the handwritten digit. Similarly to Meunier et al. (2022), we convert each image to a histogram so that the SW kernels have probability distributions as inputs. For image $i \in [N]$, let $(x_k^i, y_k^i)_{k=1}^{n_i}$ denote the coordinates of the n_i active pixels and $(\psi_k^i)_{k=1}^{n_i}$ the associated pixel intensities. The positions are renormalized to the grid $[0, 1]^2$ and we consider the weighted histogram $\mu_i = \sum_{k=1}^{n_i} \alpha_k^i \delta_{(x_k^i, y_k^i)}$ where $\alpha_k^i = \psi_k^i / \sum_{l=1}^{n_i} \psi_l^i$ are the normalized intensities. We use a train/test split of size 80/20 giving $N_{\text{train}} = 1437$ training and $N_{\text{test}} = 360$ testing images, respectively, where each set is balanced.

We implement a Kernel SVM classifier where the SW_2 distance involved in the kernel expression is either computed with the standard Monte Carlo estimate (MC) or with the SHCV estimate (SHCV). The number of random projections n evolves in $\{25; 50; 75; 100\}$. To compare different scenarios, the bandwidth parameter $\gamma > 0$ of the Gaussian kernel is in $\{0.01; 1\}$ and the regularization parameter C of the SVM is optimized with a cross-validation strategy. Table 4.2 reports the means and standard deviations of the test accuracies of the different methods where the statistics are obtained over 10 independent runs.

4.6 Theoretical analysis

This section gathers several theoretical properties of the novel SHCV estimate. We first state some elementary properties of the underlying integration

n	$\gamma = 0.01$		$\gamma = 1$	
	MC	SHCV	MC	SHCV
25	86.0 $\pm$ 0.6	87.3 $\pm$ 0.3	92.7 $\pm$ 0.6	93.3 $\pm$ 0.6
50	86.7 $\pm$ 0.5	87.6 $\pm$ 0.3	93.2 $\pm$ 0.4	94.0 $\pm$ 0.3
75	86.8 $\pm$ 0.2	87.8 $\pm$ 0.2	93.3 $\pm$ 0.4	94.3 $\pm$ 0.2
100	86.9 $\pm$ 0.3	87.7 $\pm$ 0.1	93.3 $\pm$ 0.3	94.3 $\pm$ 0.2

Table 4.2: Kernel SVM: test accuracies with means and standard deviations. The statistics are obtained over 10 independent runs.

rule with respect to a general distribution P on the sphere and then provide a convergence rate in probability of the integration error in case of the uniform distribution U on the sphere.

4.6.1 Elementary properties

By definition of the OLSMC estimate, the SHCV estimate is exact on the linear span of the control variates (Portier and Segers, 2019). Thus, if $f_{\mu,v}^{(p)}$ happens to be a polynomial function, then the SHCV estimate will be exact as soon as the maximal degree of the spherical harmonics is equal to or larger than the degree of $f_{\mu,v}^{(p)}$.

Proposition 4.2 (Exact rule). *If $f_{\mu,v}^{(p)}$ is a polynomial function with degree m , considering the SHCV estimate and control variates $\varphi = (\varphi_j)_{j=1}^{s_{L,d}}$, if $2L \geq m$ and $n > s_{L,d}$ then SHCV is exact, i.e., $\text{SHCV}_{n,L}^p(\mu, \nu) = \text{SW}_p^p(\mu, \nu)$.*

In the Gaussian case of Remark 4.1, for two proportional dispersion matrices A and B , say $B = \gamma A$ for some $\gamma > 0$, the formula of $f_{\mu,v}^{(2)}(\theta) = W_2^2(\theta_\#^* \mu, \theta_\#^* \nu)$ simplifies to a quadratic homogeneous polynomial. Therefore, the SHCV estimate is exact³ as soon as the spherical harmonics of degree two are included as control variates, since the centered integrand is then contained in the linear space spanned by the control variates. Actually, this remains true in case $p = 2$ if the measures are related by an affine transformation.

Corollary 4.1 (Affine transform). *In case $\mu, \nu \in \mathcal{P}_2(\mathbb{R}^d)$ are related by $X \sim \mu$ and $\alpha X + b \sim \nu$ where $\alpha \in (0, \infty)$ and $b \in \mathbb{R}^d$ then the SHCV estimate is exact.*

The SHCV estimator with spherical harmonics of maximal degree two or higher is guaranteed to have an asymptotic variance that is bounded above

³See numerical comparison in Section 4.8.4 where the SHCV estimate is the only one achieving exact integration (zero error).

by the lower- and upper-CV methods of Nguyen and Ho (2024). Indeed, the latter two methods are control variate methods, each based on a *single* control variate, which happens to be a quadratic polynomial.

Corollary 4.2 (Mean invariance). *For $\mu, \nu \in \mathcal{P}_2(\mathbb{R}^d)$, the error of the SHCV method is (exactly) invariant under changes of the mean vectors m_μ and m_ν of μ and ν respectively.*

This result implies that we can, without loss of generality, suppose that μ and ν are centered. The reason is that the effect of the two mean vectors being potentially different is the translation term $\langle \theta, m_Y - m_X \rangle$ coming from the optimal transport map that pushes $\theta_\#^\star \mu$ to $\theta_\#^\star \nu$, and this term produces an additional quadratic term $\langle \theta, m_Y - m_X \rangle^2$ in the function $f_{\mu, \nu}^{(2)}$. This additional quadratic polynomial in θ is integrated exactly by SHCV; its integral is $\|m_X - m_Y\|^2/d$. See Remark 4.1 for the Gaussian case.

4.6.2 Asymptotic error bound

We provide a convergence rate in probability⁴ of the integration error $\text{SHCV}_{n,L}^p(\mu, \nu) - \text{SW}_p^p(\mu, \nu)$ for general probability measures $\mu, \nu \in \mathcal{P}_p(\mathbb{R}^d)$ and $p \in [1, \infty)$, where $\text{SW}_p(\mu, \nu)$ is the SW distance with respect to the uniform distribution on the sphere and $\text{SHCV}_{n,L}^p(\mu, \nu)$ is our estimate in Definition 4.4 based on n random projections on the sphere and using spherical harmonics of even degree up to $2L$ as control variates. The guarantee relies on two properties: first, a general convergence rate of the integration error of the OLSCV estimate in Portier and Segers (2019, Theorem 1), available in Section 4.8.3, and second, a bound on the uniform approximation error of a Lipschitz function on the sphere by its orthogonal projection on the space of polynomials (Newman and Shapiro, 1964, Theorem 2). The latter result applies since the integrand $f_{\mu, \nu}^{(p)}$ in Equation (4.3) is always at least Lipschitz: by Han (2023, Theorem 2.4), we have

$$\left| f_{\mu, \nu}^{(p)}(\theta) - f_{\mu, \nu}^{(p)}(\gamma) \right| \leq M_p(\mu, \nu) \|\theta - \gamma\| \quad (4.9)$$

for $\theta, \gamma \in \mathbb{S}^{d-1}$, where $M_p(\mu, \nu)$ is an explicit function of p and the p th order moments of μ and ν .

The modulus of continuity of $f \in \mathcal{C}(\mathbb{S}^{d-1})$ is defined for $h > 0$ by

$$\omega(h) = \sup\{|f(\theta) - f(\gamma)| : \rho(\theta, \gamma) \leq h\}$$

with $\rho(\theta, \gamma) = \arccos\langle \theta, \gamma \rangle$ the length of the arc connecting $\theta, \gamma \in \mathbb{S}^{d-1}$.

⁴The notation $R_n = O_{\mathbb{P}}(a_n)$, for random variables R_n and constants $a_n > 0$, means that for every $\delta > 0$ there exists $K_\delta > 0$ such that $\mathbb{P}(|R_n| > K_\delta a_n) < \delta$ for all n .

Theorem 4.1 (Newman and Shapiro (1964), Theorem 2). *There exists an absolute constant $A > 0$ such that for each dimension $d \geq 2$, each function $f \in \mathcal{C}(\mathbb{S}^{d-1})$ and each degree L there exists a polynomial $P_{f,L}$ on $\mathbb{S}^{d-1}$ of degree at most L such that*

$$\sup_{\theta \in \mathbb{S}^{d-1}} |f(\theta) - P_{f,L}(\theta)| \leq A \omega(d/L).$$

The polynomial $P_{f,L}$ in Theorem 4.1 that yields the uniform approximation is in general not the same as the orthogonal projection $\Pi_{f,L} = \sum_{\ell=0}^L \sum_{k=1}^{N_\ell^d} \hat{f}_{\ell,k} \varphi_{\ell,k}$ of f in $L_2(\mathbb{S}^{d-1})$ on the space of spherical harmonics of degree at most L , where we use the notation of Proposition 4.1. Still, the $L_2(\mathbb{S}^{d-1})$ approximation error of $\Pi_{f,L}$ is bounded by the one of $P_{f,L}$, which is in turn bounded by the uniform approximation error of $P_{f,L}$. Applying this argument to the integrand $f_{\mu,\nu}^{(p)}$, we obtain the following convergence rate in probability of the SHCV estimate.

Theorem 4.2 (Convergence rate). *Let $d \geq 2$, $p \in [1, \infty)$ and $\mu, \nu \in \mathcal{P}_p(\mathbb{R}^d)$ be fixed. For any degree sequence $L = L_n$ such that $L = o(n^{1/(2(d-1))})$ as $n \rightarrow \infty$, the integration error of the SHCV estimate with respect to the uniform distribution on the sphere satisfies*

$$\left| \text{SHCV}_{n,L}^p(\mu, \nu) - \text{SW}_p^p(\mu, \nu) \right| = O_{\mathbb{P}}(L^{-1} n^{-1/2}). \quad (4.10)$$

The proof of Theorem 4.2 is given in Section 4.8.3. Writing $L = n^{1/(2(d-1))} / \ell_n$ where $\ell_n > 0$ diverges to infinity but can do so arbitrarily slowly (for instance logarithmically), the convergence rate in Equation (4.10) becomes

$$\left| \text{SHCV}_{n,L}^p(\mu, \nu) - \text{SW}_p^p(\mu, \nu) \right| = O_{\mathbb{P}}(\ell_n n^{-\frac{1}{2} \frac{d}{d-1}}).$$

For $d = 3$, this yields the rate $n^{-3/4+o(1)}$ for the SHCV estimate, in comparison to the Monte Carlo rate $n^{-1/2}$.

If the integrand $f_{\mu,\nu}^{(p)}$ has higher-order derivatives, then a faster convergence rate can be expected because of approximation results in Ragozin (1971) improving Theorem 4.1. This concerns for instance the SW distance between two Gaussian distributions, for which the integrand in Remark 4.1 has derivatives of all orders. In contrast, for discrete measures, the integrand will not even be continuously differentiable, and the Lipschitz property is the best we can hope for. Both cases are illustrated in Section 4.8.4 in the Appendix.

4.7 Conclusion

We have developed a novel method for reducing the variance of Monte Carlo estimation of the SW distance using spherical harmonics as control variates.

The excellent practical performance of the SHCV estimate against state-of-the-art baselines is confirmed by theoretical properties and a convergence rate in probability for the integration error.

Using control variates with QMC sequences is usually *not* implemented in the same way as with Monte Carlo sequences (Hickernell et al., 2005) and would require a particular treatment which is beyond the scope of this chapter. In statistical inference with parametric probability measures, note that SHCV is compatible with the computation of gradient $\nabla_\phi SW_p^p(\mu, \nu_\phi)$ and can be used for *generalized* SW flows (Kolouri et al., 2019). While the proposed SHCV estimate focuses on the uniform distribution on $\mathbb{S}^{d-1}$, it can be extended to more general probability distributions by combining control variates with importance sampling techniques as in Leluc et al. (2022).

4.8 Appendix

Section 4.8.1 gathers the pseudo-code of all the algorithms used in the experiments while Section 4.8.2 presents additional details and results for the numerical experiments. Section 4.8.3 gathers the proofs of the theoretical results and Section 4.8.4 is dedicated to additional results on spherical harmonics with (1) a case-study on the regularity of the integrand $f_{\mu,\nu}^{(p)}$ and (2) a numerical check of the exact integration rule.

4.8.1 Algorithms for sliced-Wasserstein estimates

This section presents algorithmic details of the different Sliced-Wasserstein estimates in the numerical experiments:

- Algorithm 4.3 corresponds to the standard Monte Carlo estimate where random projections $\theta_1, \dots, \theta_n \sim P = \mathcal{U}(\mathbb{S}^{d-1})$ are used to compute the Wasserstein distance of projected distributions with $f_{\mu,\nu}^{(p)}(\theta) = W_p^p(\theta_\#^\star \mu, \theta_\#^\star \nu)$.
- Algorithm 4.4 describes the proposed SHCV estimate which relies on the OLS procedure with spherical harmonics φ_j for $j = 1, \dots, s$ as control variates.
- Algorithm 4.5 and Algorithm 4.6 describe the two estimates of Nguyen and Ho (2024) which rely on Gaussian approximation with lower and upper control variates.
- Algorithm 4.7 presents the control neighbors technique of Leluc et al. (2024b) and consists in building a control functional estimate with the 1-nearest neighbor estimate of the integrand f .
- Algorithm 4.8 implements a (Randomized) Quasi-Monte Carlo estimate. A low-discrepancy sequence on the unit cube $[0, 1]^d$ is mapped on $\mathbb{S}^{d-1}$ to ob-

tain a deterministic QMC point set on the unit sphere. Next, rotations from the orthogonal group are applied to randomize the samples

Algorithm 4.3 Sliced Wasserstein Monte Carlo

Require: $\mu, \nu \in \mathcal{P}_p(\mathbb{R}^d)$, number of random projections n

- 1: Sample random projections $\theta_1, \dots, \theta_n$ uniformly on $\mathbb{S}^{d-1}$
 - 2: Compute $f_n = (f_{\mu, \nu}^{(p)}(\theta_i))_{i=1}^n$
 - 3: Return average $\text{MC}_n = (\mathbf{1}_n/n)^\top f_n$
-

Algorithm 4.4 Spherical Harmonics Control Variate Estimate

Require: $\mu, \nu \in \mathcal{P}_p(\mathbb{R}^d)$, number of projections n , spherical harmonics $\varphi = (\varphi_j)_{j=1}^s$

- 1: Sample random projections $\theta_1, \dots, \theta_n$ uniformly on $\mathbb{S}^{d-1}$
 - 2: Compute $f_n = (f_{\mu, \nu}^{(p)}(\theta_i))_{i=1}^n$ and $\Phi = (\varphi(\theta_i)^\top)_{i=1}^n$
 - 3: $(\text{I}_n^{\text{ols}}, \beta_n) \in \arg \min_{\alpha, \beta} \|f_n - \alpha \mathbf{1}_n - \Phi \beta\|_2^2$
 - 4: Return $\text{SHCV}_n = \text{I}_n^{\text{ols}}$
-

Algorithm 4.5 Lower-CV Sliced Wasserstein (Nguyen and Ho, 2024)

Require: $\mu = \sum_{i=1}^{m_X} \alpha_i \delta_{x_i}$ and $\nu = \sum_{j=1}^{m_Y} \beta_j \delta_{y_j}$, number of random projections n

- 1: Sample random projections $\theta_1, \dots, \theta_n$ uniformly on $\mathbb{S}^{d-1}$
 - 2: Compute $f_n = (f_{\mu, \nu}^{(p)}(\theta_i))_{i=1}^n$
 - 3: Compute $\text{MC}_n = (\mathbf{1}_n/n)^\top f_n$
 - 4: Compute $\bar{x} = \sum_{i=1}^{m_X} \alpha_i x_i$, and $\bar{y} = \sum_{j=1}^{m_Y} \beta_j y_j$
 - 5: $C = (c_1, \dots, c_n)$ with $c_k = (\theta_k^\top (\bar{x} - \bar{y}))^2$
 - 6: Compute $b = \|\bar{x} - \bar{y}\|_2^2/d$
 - 7: Compute $\gamma_{\text{low}} = \frac{\langle f_n - \text{MC}_n, C - b \rangle}{\|C - b\|_2^2}$ and control variate $C = (\mathbf{1}_n^\top (C - b))/n$
 - 8: Return $\text{CV}_{\text{low}} = \text{MC}_n - \gamma_{\text{low}} C$
-

Algorithm 4.6 Upper-CV Sliced-Wasserstein (Nguyen and Ho, 2024)

Require: $\mu = \sum_{i=1}^{m_X} \alpha_i \delta_{x_i}$ and $\nu = \sum_{j=1}^{m_Y} \beta_j \delta_{y_j}$, number of random projections n

- 1: Sample random projections $\theta_1, \dots, \theta_n$ uniformly on $\mathbb{S}^{d-1}$
- 2: Compute $f_n = (f_{\mu, \nu}^{(p)}(\theta_i))_{i=1}^n$
- 3: Compute $\text{MC}_n = (\mathbf{1}_n/n)^\top f_n$
- 4: Compute $\bar{x} = \sum_{i=1}^{m_X} \alpha_i x_i$, and $\bar{y} = \sum_{j=1}^{m_Y} \beta_j y_j$
- 5: $C = (c_1, \dots, c_n)$ with $c_k = (\theta_k^\top (\bar{x} - \bar{y}))^2 + \sum_{i=1}^{m_X} \alpha_i (\theta_k^\top (x_i - \bar{x}))^2 + \sum_{j=1}^{m_Y} \beta_j (\theta_k^\top (y_j - \bar{y}))^2$
- 6: Compute $b = \left(\|\bar{x} - \bar{y}\|_2^2 + \sum_{i=1}^{m_X} \alpha_i \|x_i - \bar{x}\|_2^2 + \sum_{j=1}^{m_Y} \beta_j \|y_j - \bar{y}\|_2^2 \right) / d$
- 7: Compute $\gamma_{\text{up}} = \frac{\langle f_n - \text{MC}_n, C - b \rangle}{\|C - b\|_2^2}$ and control variate $C = (\mathbf{1}_n^\top (C - b)) / n$
- 8: Return $\text{CV}_{\text{up}} = \text{MC}_n - \gamma_{\text{up}} C$

Algorithm 4.7 Control Neighbors Sliced Wasserstein Leluc et al. (2024b)

Require: $\mu, \nu \in \mathcal{P}_p(\mathbb{R}^d)$, number of random projections n , probability $P \in \mathcal{P}(\mathbb{S}^{d-1})$

- 1: Sample random projections $\theta_1, \dots, \theta_n \sim P$
- 2: Compute $f_n = (f_{\mu, \nu}^{(p)}(\theta_i))_{i=1}^n$
- 3: Compute $\text{MC}_n = (\mathbf{1}_n/n)^\top f_n$
- 4: Compute nearest neighbor evaluations $\hat{f}_n = (\hat{f}_n^{(1)}(\theta_1), \dots, \hat{f}_n^{(n)}(\theta_n))$
- 5: Return $\text{CVNN}_n = \text{MC}_n - [\mathbf{1}_n^\top \{\hat{f}_n - \mathbf{I}(\hat{f}_n)\}] / n$

Algorithm 4.8 Randomized Quasi-Monte Carlo Sliced Wasserstein (Nguyen et al., 2024)

Require: $\mu, \nu \in \mathcal{P}_p(\mathbb{R}^d)$, number of projections n , Φ the cdf of $\mathcal{N}(0, 1)$

- 1: Generate low-discrepancy sequence $U_1, \dots, U_n$ on $[0, 1]^d$
- 2: Compute Gaussian mapping $v_i = \Phi^{-1}(U_i) / \|\Phi^{-1}(U_i)\|_2$, where the standard normal quantile function Φ^{-1} is applied component-wise.
- 3: Generate a random rotation $R \sim \mathcal{U}(O_d(\mathbb{R}))$ where $O_d(\mathbb{R}) = \{R \in \mathbb{R}^{d \times d} \mid R^\top R = I_d\}$
- 4: Generate random projections $\theta_i = R v_i$
- 5: Compute $f_n = (f_{\mu, \nu}^{(p)}(\theta_i))_{i=1}^n$
- 6: Return $\text{RQMC}_n = (\mathbf{1}_n/n)^\top f_n$

Remarks and practical details on the algorithms.

- In Algorithm 4.5, with the notation $z = \bar{x} - \bar{y}$, the control variate is the centered version of the function $\theta \mapsto |\theta^\top z|^2$, which has known integral $\int_{\mathbb{S}^{d-1}} |\theta^\top z|^2 dU(\theta) = \|z\|^2 / d = b$ (Step 6 in the algorithm) and where U de-

notes the uniform distribution on $\mathbb{S}^{d-1}$. The Lower-CV estimate is thus an OLSMC estimate with control variate $\varphi(\theta) = |\theta^\top z|^2 - \|z\|^2/d$, a polynomial in θ of degree two.

- Algorithm 4.6 is actually an OLSMC estimate with a single control variate given by the centered version of $\theta \mapsto \theta^\top M \theta$ where $M = M_0 + M_X + M_Y$ with $M_0 = (\bar{x} - \bar{y})(\bar{x} - \bar{y})^\top$, $M_X = \sum_{i=1}^{m_X} \alpha_i (x_i - \bar{x})(x_i - \bar{x})^\top$ and similarly for M_Y . The integral with respect to the uniform distribution on the sphere is $\int_{\mathbb{S}^{d-1}} \theta^\top M \theta dU(\theta) = \text{Tr}(M)/d = b$ in Step 6 of the algorithm. The control variate is thus the centered quadratic polynomial $\varphi(\theta) = \theta^\top M \theta - \text{Tr}(M)/d$.
- In Step 4 of Algorithm 4.7, $\hat{f}_n^{(i)}(\theta_i)$ is a leave-one-out nearest-neighbor estimate computed with the Euclidean metric. In Step 5, computing $I(\hat{f}_n)$ requires sampling M additional directions θ_i from $\mathcal{P}$ and we set $M = n^{1+2/d}$ as recommended by the theory in Leluc et al. (2024b).
- In Algorithm 4.8, random orthogonal matrices $R \in O_d(\mathbb{R})$ are generated using the function `ortho_group` of the Python package `scipy` (Virtanen et al., 2020). This function returns random orthogonal matrices drawn from the Haar distribution using a careful QR decomposition⁵ as in Mezzadri (2007).

4.8.2 Numerical details and additional results

To assess the finite-sample performance of the SHCV estimate, we first present some synthetic data examples involving the computation of $\text{SW}_2^2(\mu, \nu)$ between multivariate Gaussian distributions. We consider different dimensions $d \in \{3; 5; 6; 10; 20\}$ and a varying number of random projections $n \in [10^2; 10^4]$. We consider the two cases of *exact* distributions and *sampled* distributions. Then, we focus on the Sliced-Wasserstein distance $\text{SW}_2(\mu_m, \nu_m)$ between 3D point clouds from the ShapeNetCore dataset of Chang et al. (2015). In both cases, the true value of the Sliced-Wasserstein distance is estimated with a very large number $n = 10^8$ of projections. The different results report the mean squared error $\mathbb{E}[|\text{SW}_{2,n}^2(\mu, \nu) - \text{SW}_2^2(\mu, \nu)|^2]$ for the different estimates $\text{SW}_{2,n}^2(\mu, \nu)$ where the expectation is computed as an average over 100 independent runs. The experiments were performed on a laptop Intel Core i7-10510U CPU 1.80GHz $\times$ 8.

Methods in competition. Along with the standard Monte Carlo estimate (Algorithm 4.3) and the proposed SHCV estimate (Algorithm 4.4), we consider three other baselines based on control variates: the two estimates of Nguyen and Ho (2024) based on lower and upper bounds of a Gaussian approximation (Algorithm 4.5 and Algorithm 4.6) and the *control neighbors* estimate of Leluc

⁵QR decomposition refers to the factorization of a matrix X into a product $X = QR$ of an orthonormal matrix Q and an upper triangular matrix R .

et al. (2024b) based on nearest neighbors estimates acting as control variates (Algorithm 4.7). For completeness, we also include a comparison with (Randomized) Quasi Monte Carlo procedures as in Nguyen et al. (2024) where deterministic QMC sets from $[0, 1]^d$ are mapped to the unit sphere and randomly rotated (Algorithm 4.8). The methods in competition are: the standard Monte Carlo estimate (MC), the lower-CV (CV_{low}) and upper-CV (CV_{up}) estimates, the control neighbors estimate (CVNN), the proposed estimate with spherical harmonics as control variates (SHCV), the Quasi Monte Carlo (QMC) and Random Quasi Monte Carlo estimates (RQMC).

Multivariate Gaussian: exact case with distributions μ, ν and smooth integrands

We compute $SW_2^2(\mu, \nu)$ between $\mu = \mathcal{N}(a, \mathbf{A})$ and $\nu = \mathcal{N}(b, \mathbf{B})$ with means $a, b \sim \mathcal{N}(\mathbf{1}_d, I_d)$ and covariance matrices $\mathbf{A} = \Sigma_a \Sigma_a^\top$ and $\mathbf{B} = \Sigma_b \Sigma_b^\top$ where all the entries of Σ_a, Σ_b are drawn according to $\mathcal{N}(0, 1)$. The integrand of interest is $f_{\mu, \nu}^{(2)}(\theta) = |\theta^\top(a - b)|^2 + (\sqrt{\theta^\top \mathbf{A} \theta} - \sqrt{\theta^\top \mathbf{B} \theta})^2$. Figure 4.4 reports the MSE of the different methods with respect to both the number of projections and the computing time in dimension $d \in \{3; 6\}$, while Table 4.3 reports the MSE and computing times (in ms) in dimension $d \in \{5; 10; 20\}$.

Method	$d = 5$		$d = 10$		$d = 20$	
	MSE	Time	MSE	Time	MSE	Time
MC	2.6e-2	06.4 $\pm$ 1.1	8.4e-3	06.7 $\pm$ 0.8	2.0e-2	06.6 $\pm$ 0.5
CV_{low}	4.7e-3	11.1 $\pm$ 3.4	3.0e-3	11.6 $\pm$ 3.0	4.4e-3	12.1 $\pm$ 2.6
CV_{up}	1.5e-2	11.4 $\pm$ 3.5	7.0e-3	11.8 $\pm$ 3.1	1.6e-2	13.2 $\pm$ 2.8
CVNN	7.7e-3	43.4 $\pm$ 13	1.0e-2	60.0 $\pm$ 16	2.6e-2	70.6 $\pm$ 14
QMC	4.9e-3	22.4 $\pm$ 3.4	4.1e-3	41.1 $\pm$ 11	1.3e-2	60.9 $\pm$ 7.1
RQMC	7.5e-4	25.6 $\pm$ 6.8	2.4e-3	40.9 $\pm$ 11	1.4e-2	66.4 $\pm$ 14
SHCV	4.8e-5	09.8 $\pm$ 5.1	1.7e-3	08.9 $\pm$ 1.2	2.7e-3	14.3 $\pm$ 2.0

Table 4.3: Mean Squared Error (MSE) and computing time (ms) for Gaussian distributions in dimension $d \in \{5; 10; 20\}$ and $n = 500$.

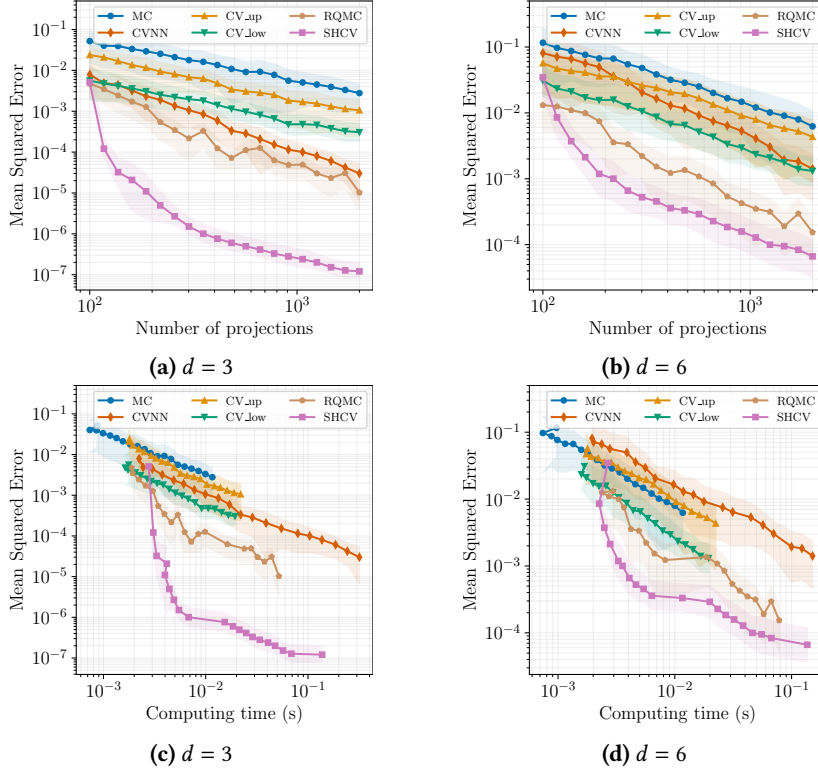


Figure 4.4: MSE and computing time for Gaussian distributions, dimension $d \in \{3; 6\}$, obtained over 100 replications.

Multivariate Gaussian: sampled case with discrete distributions

μ_m, ν_m

We compute $\text{SW}_2^2(\mu_m, \nu_m)$ between empirical distributions $\mu_m = m^{-1} \sum_{i=1}^m \delta_{x_i}$ and $\nu_m = m^{-1} \sum_{j=1}^m \delta_{y_j}$,

$$x_1, \dots, x_m \sim \mu = \mathcal{N}(a, \mathbf{A}), \quad y_1, \dots, y_m \sim \nu = \mathcal{N}(b, \mathbf{B})$$

with means $a, b \sim \mathcal{N}_d(\mathbf{1}_d, I_d)$ and covariance matrices $\mathbf{A} = \Sigma_a \Sigma_a^\top$ and $\mathbf{B} = \Sigma_b \Sigma_b^\top$ where all the entries of Σ_a, Σ_b are drawn according to $\mathcal{N}(0, 1)$. This time, instead of evaluating the integrand $f_{\mu, \nu}^{(2)}$ directly, we assume that one only has access to data samples from the source and the target. We use $m = 1000$ data samples from the source $x_1, \dots, x_m \sim \mu$ and the target $y_1, \dots, y_m \sim \nu$. Figure 4.5 reports the mean squared errors of the different methods with respect to both the number of projections and the computing time in dimension $d \in \{3; 6\}$. Tables 4.4 and 4.5 report the MSE and computing times (in ms) in dimension $d \in \{5; 10; 20\}$.

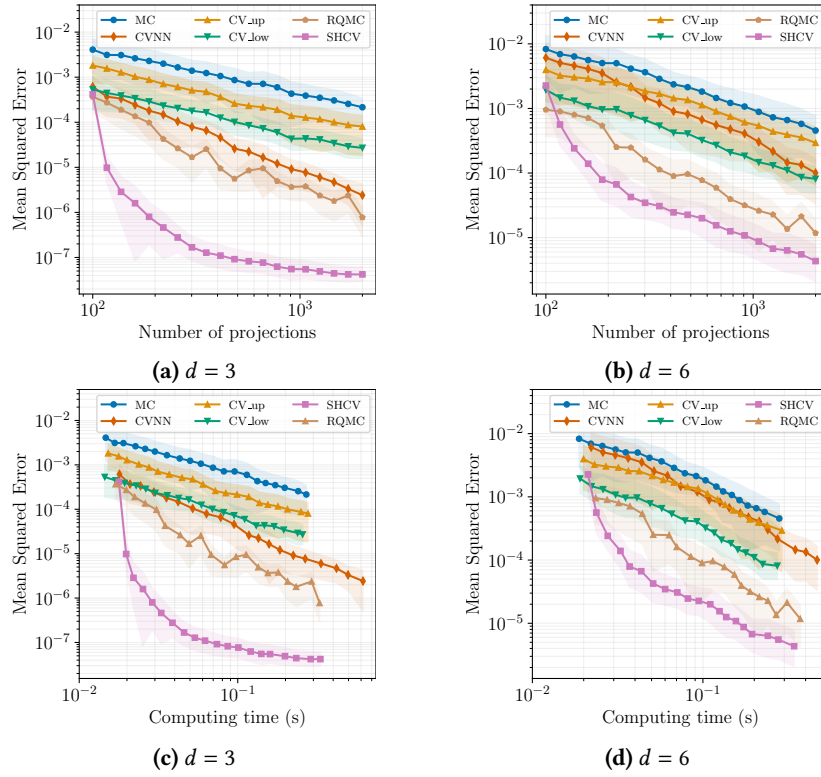


Figure 4.5: MSE for sampled Gaussian distributions supported on $m = 1000$ points, dimension $d \in \{3, 6\}$, obtained over 100 replications.

Method	$d = 5$		$d = 10$		$d = 20$	
	$n = 500$	$n = 1000$	$n = 500$	$n = 1000$	$n = 500$	$n = 1000$
MC	1.45e-3	8.05e-4	9.45e-4	5.46e-4	1.47e-3	6.67e-4
CV _{low}	2.67e-4	1.46e-4	3.45e-4	1.63e-4	3.82e-4	1.73e-4
CV _{up}	8.44e-4	4.82e-4	7.51e-4	4.76e-4	1.09e-3	4.73e-4
CVNN	4.29e-4	1.79e-4	1.12e-3	4.77e-4	2.14e-3	1.44e-3
QMC	2.91e-4	7.97e-5	2.37e-4	8.35e-5	6.60e-4	1.42e-4
RQMC	5.80e-5	1.91e-5	2.75e-4	1.20e-4	1.17e-3	3.64e-4
SHCV	2.68e-6	5.63e-7	1.93e-4	6.86e-5	2.95e-4	1.16e-4

Table 4.4: Mean Squared Error (MSE) for sampled Gaussian distributions supported on $m = 1000$ points in dimension $d \in \{5; 10; 20\}$ with $n \in \{500; 1000\}$ projections, obtained over 100 independent runs.

Method	$d = 5$		$d = 10$		$d = 20$	
	$n = 500$	$n = 1000$	$n = 500$	$n = 1000$	$n = 500$	$n = 1000$
MC	81.1 ± 3.5	152 ± 0.9	80.7 ± 4.4	152 ± 1.2	81.1 ± 1.8	155 ± 2.3
CV _{low}	79.7 ± 1.1	151 ± 1.1	80.1 ± 1.4	150 ± 2.2	80.0 ± 1.0	155 ± 4.0
CV _{up}	83.0 ± 1.2	157 ± 1.7	83.0 ± 1.7	157 ± 1.6	83.1 ± 1.5	160 ± 3.0
CVNN	110 ± 2.2	208 ± 1.5	122 ± 1.6	244 ± 2.1	127 ± 1.4	275 ± 3.6
QMC	100 ± 1.2	174 ± 1.3	113 ± 1.4	187 ± 1.0	129 ± 1.4	214 ± 3.5
RQMC	96.3 ± 2.2	190 ± 1.4	113 ± 1.2	188 ± 2.4	130 ± 1.0	214 ± 3.8
SHCV	89.0 ± 6.3	175 ± 3.4	89.0 ± 4.5	176 ± 2.4	88.1 ± 2.8	180 ± 5.6

Table 4.5: Computing time (ms) for sampled Gaussian distributions supported on $m = 1000$ points in dimension $d \in \{5; 10; 20\}$ with $n \in \{500; 1000\}$ projections, obtained over 100 independent runs.

Empirical distributions of 3D point clouds

We use three point clouds taken at random from the ShapeNetCore dataset of Chang et al. (2015) corresponding to the objects plane, lamp and bed, each composed of $m = 2048$ points of $\mathbb{R}^3$ (see Figure 4.6 below for visualization). The SHCV estimate uses spherical harmonics up to degree $2L = 16$ leading to $s = 152$ control variates. Figure 4.3 reports the boxplots of the error distribution $(\widehat{SW}_n(\mu_m, \nu_m) - SW(\mu_m, \nu_m))$ for the different SW estimates based on n random projections with $n \in \{100; 250; 500; 1000\}$ obtained over 100 independent runs while Table 4.6 presents the mean squared errors $E[|\widehat{SW}_n(\mu_m, \nu_m) - SW(\mu_m, \nu_m)|^2]$ for the different SW estimates with $n \in \{100; 500; 1000\}$. Once again, our SHCV estimate provides the best performance with huge gains in terms of variance reduction.

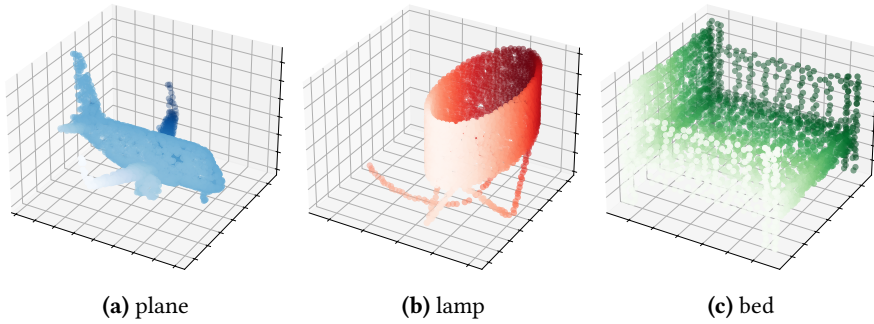


Figure 4.6: Visualization of 3D point clouds of plane, lamp, bed.

Method	PLANE/LAMP			LAMP/BED			PLANE/BED		
	$n = 100$	$n = 500$	$n = 1000$	$n = 100$	$n = 500$	$n = 1000$	$n = 100$	$n = 500$	$n = 1000$
MC	5.1e-5	1.1e-5	6.0e-6	2.0e-4	3.8e-5	1.5e-5	1.3e-4	2.6e-5	1.2e-5
CV _{low}	4.8e-5	9.3e-6	5.1e-6	9.3e-5	1.4e-5	6.5e-6	4.4e-5	9.8e-6	5.2e-6
CV _{up}	1.1e-5	2.2e-6	1.3e-6	1.1e-5	2.3e-6	1.0e-6	6.6e-5	1.5e-5	8.7e-6
CVNN	1.3e-5	5.2e-7	1.8e-7	2.4e-5	1.1e-6	3.5e-7	3.2e-5	1.1e-6	3.3e-7
QMC	1.4e-5	6.9e-7	2.8e-8	2.4e-5	3.8e-7	2.9e-7	1.2e-4	1.2e-7	4.9e-7
RQMC	7.7e-6	2.2e-7	8.1e-8	1.5e-5	2.7e-7	1.3e-7	2.6e-5	1.0e-6	3.0e-7
SHCV	5.3e-6	2.1e-8	2.1e-8	2.5e-6	1.5e-8	1.4e-8	1.3e-5	3.7e-8	3.6e-8

Table 4.6: Mean Squared Error $E[|\widehat{SW}_n(\mu_m, v_m) - SW(\mu_m, v_m)|^2]$ for different SW estimates based on n random projections with $n \in \{100; 500; 1000\}$ obtained over 100 independent runs.

Method	PLANE/LAMP			LAMP/BED			PLANE/BED		
	$n = 100$	$n = 500$	$n = 1000$	$n = 100$	$n = 500$	$n = 1000$	$n = 100$	$n = 500$	$n = 1000$
MC	36.4 $\pm$ 10	184 $\pm$ 3	309 $\pm$ 3	37.9 $\pm$ 10	184 $\pm$ 3	308 $\pm$ 3	35.6 $\pm$ 10	185 $\pm$ 3	308 $\pm$ 3
CV _{low}	36.8 $\pm$ 10	180 $\pm$ 1	301 $\pm$ 1	38.4 $\pm$ 10	181 $\pm$ 2	300 $\pm$ 2	36.1 $\pm$ 11	181 $\pm$ 1	300 $\pm$ 1
CV _{up}	38.1 $\pm$ 10	187 $\pm$ 2	316 $\pm$ 1	39.7 $\pm$ 10	187 $\pm$ 1	315 $\pm$ 1	37.2 $\pm$ 11	188 $\pm$ 1	314 $\pm$ 1
CVNN	40.4 $\pm$ 11	220 $\pm$ 1	417 $\pm$ 1	42.0 $\pm$ 11	220 $\pm$ 1	414 $\pm$ 2	39.5 $\pm$ 11	221 $\pm$ 1	414 $\pm$ 1
QMC	32.9 $\pm$ 12	194 $\pm$ 3	324 $\pm$ 1	27.9 $\pm$ 10	194 $\pm$ 2	327 $\pm$ 1	38.0 $\pm$ 13	195 $\pm$ 2	325 $\pm$ 1
RQMC	40.6 $\pm$ 11	202 $\pm$ 2	346 $\pm$ 2	42.0 $\pm$ 10	203 $\pm$ 4	344 $\pm$ 2	39.3 $\pm$ 11	202 $\pm$ 1	343 $\pm$ 1
SHCV	37.9 $\pm$ 10	185 $\pm$ 2	311 $\pm$ 1	39.7 $\pm$ 10	185 $\pm$ 1	310 $\pm$ 1	37.1 $\pm$ 11	186 $\pm$ 1	310 $\pm$ 1

Table 4.7: Computing times (ms) for different SW estimates based on n random projections with $n \in \{100; 500; 1000\}$ obtained over 100 independent runs.

Kernel Support Vector Machines and sliced-Wasserstein distances

Kernel Support Vector Machines. Kernel Support Vector Machines (SVMs) are a powerful class of supervised learning algorithms used for classification and regression tasks. At their core, SVMs aim to find a hyperplane that best separates the data into different classes while maximizing the margin between these classes. However, in cases where the classes are not linearly separable, Kernel SVMs come into play. The key idea behind Kernel SVMs is to map the input data $x \in \mathbb{R}^q$ into a higher-dimensional space via $\phi(x)$, where a linear separation might be achievable (see Figure 4.7 below).

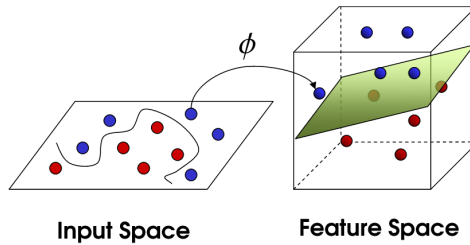


Figure 4.7: Visualization of Kernel transformation.

The kernel function $k(\cdot, \cdot)$ computes the similarity between pairs of data points in the higher-dimensional space without explicitly calculating the transformed data. Given training data $x_i \in \mathbb{R}^q$ with $i = 1, \dots, n$ divided into two classes and a vector of labels $y \in \{-1; +1\}$, the goal is to find the parameters $\omega \in \mathbb{R}^q$ and $b \in \mathbb{R}$ such that the prediction given by $\text{sign}(\omega^\top \phi(x_i) + b)$ is correct for most samples.

In Scikit-learn (Pedregosa et al., 2011), the ‘SVC class’ provides an implementation of Kernel SVMs. The objective function (also known as the loss function) for the SVM is derived from the problem of finding the optimal hyperplane that maximizes the margin between different classes. The formulation includes a regularization term C to balance the desire for a large margin with the goal of minimizing classification errors and the optimization problem of interest is given by

$$\min_{\omega, b, \zeta} \frac{1}{2} \omega^\top \omega + C \sum_{i=1}^n \zeta_i$$

$$\text{subject to } y_i (\omega^\top \phi(x_i) + b) \geq 1 - \zeta_i \text{ and } \zeta_i \geq 0, i = 1, \dots, n.$$

The parameter C , which is shared among all Support Vector Machine (SVM) kernels, balances the compromise between misclassifying training examples and maintaining a simple decision surface. A lower C value results in a smoother decision surface, while a higher C value strives to accurately classify all training examples. This problem is usually solved with the dual formulation through the Kernel matrix $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$.

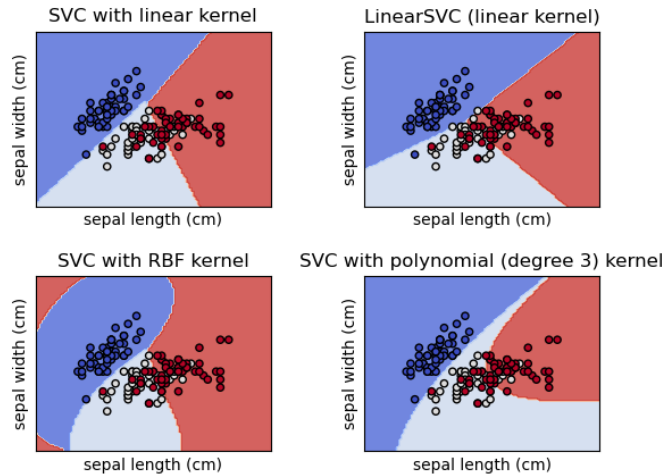


Figure 4.8: Comparison of different linear SVM classifiers on a 2D projection of the iris dataset.

Sliced-Wasserstein Gaussian kernels. SW distances inherit topological properties from the original Wasserstein distance such as the weak convergence of probability measures (Nadjahi et al., 2020b). SW distances are useful in the context of distribution regression (Szabó et al., 2016) where the goal is to learn a real-valued function defined on the space of probability distributions from a sequence of observations. The classical Wasserstein distance cannot be used to build a positive definite kernel on the space of distributions as the induced Wasserstein space has curvature (Feragen et al., 2015). To overcome this problem, a SW-based kernel for absolutely continuous distributions was first introduced in Kolouri et al. (2016) and later on extended to empirical distributions by Meunier et al. (2022). For any $\gamma > 0$, both $e^{-\gamma \text{SW}_2^2(\cdot, \cdot)}$ and $e^{-\gamma \text{SW}_1(\cdot, \cdot)}$ are valid kernels on probability measures.

Using the SW kernel of Kolouri et al. (2016) defined by $k(\mu_i, \mu_j) = \exp(-\gamma \text{SW}_2^2(\mu_i, \mu_j))$ with $\gamma > 0$, we consider the image classification task on the digits dataset of Alpaydin and Kaynak (1998). This dataset contains $N = 1797$ images of size 8×8 and the goal is to predict the label $z \in \{0, 1, \dots, 9\}$ of the handwritten digit. Similarly to Meunier et al. (2022), we convert each image to an histogram so that the SW kernels have probability distributions as inputs. We use a train/test split of size 80/20 giving $N_{\text{train}} = 1437$ training and $N_{\text{test}} = 360$ testing images respectively where each set is balanced. The Kernel SVM classification requires to compute the Gram matrices $K = (k(\mu_i, \mu_j))$ with $(i, j) \in [N_{\text{train}}] \times [N_{\text{train}}]$ and with $(i, j) \in [N_{\text{train}}] \times [N_{\text{test}}]$. Computational benefits occur with the SHCV estimate when many SW distances need to be computed. Using the projections $\theta_1, \dots, \theta_n \sim P$ and the spherical harmonics $\varphi = (\varphi_j)_{j=1}^s$ to compute $\Pi = \Phi(\Phi^\top \Phi)^{-1} \Phi^\top$ with $\Phi = (\varphi(\theta_i)^\top)_{i=1}^n$, we need to compute the weights $w = [(I - \Pi)\mathbf{1}_n] / [\mathbf{1}_n^\top (I - \Pi)\mathbf{1}_n]$ only once in order to obtain a linear integration rule $w_n^\top f_n^{(i,j)}$ with $f_n^{(i,j)} = (f_{\mu_i, v_j}^{(p)}(\theta_1), \dots, f_{\mu_i, v_j}^{(p)}(\theta_n))^\top$ to estimate $\text{SW}_p^p(\mu_i, \nu_j)$.

Algorithm 4.9 SW-Kernel Monte Carlo

Require: $\mathcal{D}_X = (\mu_1, \dots, \mu_{N_X})$, $\mathcal{D}_Y = (\nu_1, \dots, \nu_{N_Y})$, number of random projections $n \in \mathbb{N}$, $P \in \mathcal{P}(\mathbb{S}^{d-1})$

- 1: Initialize Kernel matrix $K \in \mathbb{R}^{N_X \times N_Y}$
 - 2: Sample random projections $\theta_1, \dots, \theta_n \sim P$
 - 3: **For** $i = 1, \dots, N_X$ **do**
 - 4: **For** $j = 1, \dots, N_Y$ **do**
 - 5: $f_n^{(i,j)} = (f_{\mu_i, \nu_j}^{(2)}(\theta_k))_{k=1}^n$
 - 6: $\text{MC}_n = (\mathbf{1}_n/n)^\top f_n^{(i,j)}$
 - 7: $K[i, j] = \exp(-\gamma \text{MC}_n)$
 - 8: **Return** Kernel matrix K
-

Algorithm 4.10 SW-Kernel SHCV

Require: $\mathcal{D}_X = (\mu_1, \dots, \mu_{N_X}), \mathcal{D}_Y = (v_1, \dots, v_{N_Y}), n \in \mathbb{N}, P \in \mathcal{P}(\mathbb{S}^{d-1})$,
control variates $\varphi = (\varphi_j)_{j=1}^s$

- 1: Sample random projections $\theta_1, \dots, \theta_n \sim P$
- 2: Compute $\Phi = (\varphi(\theta_i)^\top)_{i=1}^n$ and SHCV weights w_n
- 3: **For** $i = 1, \dots, N_X$ **do**
- 4: **For** $j = 1, \dots, N_Y$ **do**
- 5: $f_n^{(i,j)} = (f_{\mu_i, v_j}^{(2)}(\theta_k))_{k=1}^n$
- 6: SHCV $w_n = w_n^\top f_n^{(i,j)}$
- 7: $K[i, j] = \exp(-\gamma \text{SHCV}_n)$
- 8: **Return** Kernel matrix K

As mentioned in Remark 4.5, computational benefits occur when $K > 1$ integrals need to be computed since the SHCV time complexity scales as $O(Kn\omega_f + \omega(\Phi))$ compared to $O(Kn\omega_f)$ for the standard Monte Carlo estimate when dealing with K integrals. Thus, when K is large, the extra term $\omega(\Phi)$ becomes negligible. In the experiments we are dealing with images and probability measures $\mu_1, \dots, \mu_{N_X}, v_1, \dots, v_{N_Y}$ in dimension $d = 2$ with a number of Monte Carlo projections $n \in \{25; 50; 75; 100\}$. The total number of integrals to be computed are:

- $K_{XX} = \binom{N_X}{2} = 1437 \times 1436 / 2 = 1,024,581$ for the 'train/train' kernel matrix.
- $K_{XY} = N_X \times N_Y = 1437 \times 360 = 517,320$ for the 'train/test' kernel matrix.

Below, we compare the time (in μs) to compute the evaluations $f_n^{(i,j)} = (f_{\mu_i, v_j}^{(2)}(\theta_k))_{k=1}^n$ for a single pair (i, j) vs. the time to compute the SHCV weights w_n . These weights need to be computed only once for all pairs of images. Interestingly, this extra computing time is of the same order as the time required for the evaluations of the integrands so that in the end it becomes negligible as the SHCV time complexity scales as $O(Kn\omega_f + \omega(\Phi))$ compared to $O(Kn\omega_f)$ for the standard Monte Carlo estimate.

MC projections n	evaluations $f_n^{(i,j)}$ in $O(n\omega_f)$	SHCV weights w_n in $O(\omega(\Phi))$
25	422 $\pm$ 36	207 $\pm$ 243
50	700 $\pm$ 16	479 $\pm$ 516
75	945 $\pm$ 17	761 $\pm$ 559
100	1182 $\pm$ 20	1057 $\pm$ 654

Table 4.8: Comparison of time computations (in μs) for the evaluations of the integrands vs. SHCV weights.

4.8.3 Mathematical proofs

Proofs of Lemma 4.1 and Corollary 4.1

Proof of Lemma 4.1. Let $\mathcal{P}_\ell^d$ be the space of homogeneous polynomials of degree ℓ . Then $\dim \mathcal{P}_\ell^d = \binom{\ell+d-1}{d-1}$, the number of multi-indices of the form $(\alpha_1, \dots, \alpha_d)$ for nonnegative integer α_j such that $\sum_{k=1}^d \alpha_k = \ell$. From this space, we need to remove the polynomials that are a multiple of $x_1^2 + \dots + x_d^2$, since, on the sphere, this factor is trivial. Any such multiple is of the form $(x_1^2 + \dots + x_d^2)p(x)$ with $p \in \mathcal{P}_{\ell-2}^d$. For $\ell \geq 2$, we have (Atkinson and Han, 2012, Proposition 2.16) $N_\ell^d = \dim \mathcal{P}_\ell^d - \dim \mathcal{P}_{\ell-2}^d$. But then we have

$$\sum_{\ell=2,4,\dots,2k} N_\ell^d = \sum_{\ell=2,4,\dots,2k} \left[\binom{\ell+d-1}{d-1} - \binom{\ell-2+d-1}{d-1} \right] = \binom{2k+d-1}{d-1} - 1.$$

Proof of Corollary 4.1. For $\theta \in \mathbb{S}^{d-1}$, we have $\langle \theta, X \rangle \sim \theta_\#^\star \mu$ and $\alpha \langle \theta, X \rangle + \langle \theta, b \rangle \sim \theta_\#^\star \nu$. The optimal transport map between $\theta_\#^\star \mu$ and $\theta_\#^\star \nu$ is $x \mapsto \alpha x + \langle \theta, b \rangle$ and $f_{\mu,\nu}^{(2)}(\theta) = \mathbb{E}_\mu[\langle \theta, (1-\alpha)X - b \rangle^2]$, which is a quadratic polynomial in θ .

Asymptotic bound for OLSMC

Consider the general set-up of Section 4.3, in particular the OLSMC estimate in Equation (4.6).

Theorem 4.3 (Portier and Segers (2019)). *Let $(\Theta, \mathcal{F}, P)$ be a probability space. Let $\varphi = (\varphi_1, \dots, \varphi_s)^\top \in (L_2(P))^s$ be a vector of orthonormal control variates, where $s = s_n$ depends on the Monte Carlo sample size n . Define $q_n(\theta) = \varphi(\theta)^\top \varphi(\theta)$ for $\theta \in \Theta$. If $\sup_{\theta \in \Theta} q(\theta) = o(n/s)$ as $n \rightarrow \infty$, then $I_n^{\text{ols}}(f) - I(f) = O_{\mathbb{P}}(\sigma_n/\sqrt{n})$ as $n \rightarrow \infty$, where $\sigma_n^2 = I(\varepsilon_n^2)$, $\varepsilon_n = f - \hat{f}_s$, and $\hat{f}_s = \mathbb{E}[f] + \sum_{i=1}^s \langle f, \varphi_i \rangle \varphi_i$.*

Proof of Theorem 4.2

To apply Theorem 4.3, we consider $q_n(\theta) = \varphi(\theta)^\top \varphi(\theta)$ for $\theta \in \mathbb{S}^{d-1}$. Since $\varphi_1, \dots, \varphi_s$ are the orthonormal control variates,

$$q_n(\theta) = \sum_{i=1}^s \varphi_i^2(\theta) = \sum_{\ell=2,4,\dots,2L} \sum_{k=1}^{N_\ell^d} \varphi_{\ell,k}^2(\theta).$$

As a corollary to Theorem 2.9 in Atkinson and Han (2012), we have

$\sum_{k=1}^{N_\ell^d} \varphi_{\ell,k}^2(\theta) = N_\ell^d / |\mathbb{S}^{d-1}|$. In view of Lemma 4.1,

$$q_n(\theta) = |\mathbb{S}^{d-1}|^{-1} \sum_{\ell=2,4,\dots,2L} N_\ell^d = |\mathbb{S}^{d-1}|^{-1} s_{L,d} \quad (4.11)$$

with $s = s_{L,d} = \binom{2L+d-1}{d-1} - 1$. Applying Stirling's formula gives $s_{L,d} = O(L^{d-1})$ as $L \rightarrow \infty$. Hence the condition $\sup_{\theta \in \mathbb{S}^{d-1}} q(\theta) = o(n/s)$ in Theorem 4.3 becomes $s_{L,d}^2 = o(n)$, which is fulfilled as soon as

$$L = o\left(n^{1/(2(d-1))}\right), \quad n \rightarrow \infty. \quad (4.12)$$

Since the function $\theta \mapsto f(\theta) = f_{\mu,\nu}^{(p)}(\theta) = W_p^p(\theta_\#^* \mu, \theta_\#^* \nu)$ on $\mathbb{S}^{d-1}$ is Lipschitz (Han, 2023, Theorem 2.4), the modulus of continuity of f evaluated at $h = d/(2L)$ satisfies

$$\omega(d/(2L)) \leq \text{Lip}_\rho(f) d/(2L) \leq \text{Lip}_{\|\cdot\|}(f) d/(2L), \quad (4.13)$$

where $\text{Lip}_\rho(f) = \sup_{\theta, \gamma \in \mathbb{S}^{d-1}} |f(\theta) - f(\gamma)| / \rho(\theta, \gamma)$ and $\text{Lip}_{\|\cdot\|}(f) = \sup_{\theta, \gamma \in \mathbb{S}^{d-1}} |f(\theta) - f(\gamma)| / \|\theta - \gamma\|$. More precisely, from Theorem 2.4 in Han (2023), we know that

$$\begin{aligned} \text{Lip}_{\|\cdot\|}(f) &\leq M_p(\mu, \nu) = p 2^{p-1} \max\{M_p(\mu), M_p(\nu)\}^{(p-1)/p} \\ &\quad \cdot \left((M_p(\mu))^{1/p} + (M_p(\nu))^{1/p} \right) \end{aligned} \quad (4.14)$$

with $M_p(\mu) = \int_{\mathbb{R}^d} \|x\|^p d\mu(x)$ and $M_p(\nu) = \int_{\mathbb{R}^d} \|x\|^p d\nu(x)$. Hence, by Theorem 4.1, the residual function $\varepsilon_n = f - \Pi_{f,L}$ with $\Pi_{f,L} = \sum_{\ell=0}^L \sum_{k=1}^{N_{2\ell}^d} \langle f, \varphi_{2\ell,k} \rangle \varphi_{2\ell,k}$ satisfies

$$\sigma_n = \|\varepsilon_n\|_2 \leq \|f - P_{f,L}\|_2 \leq \|f - P_{f,L}\|_\infty \leq \frac{AM_p(\mu, \nu)d}{2L}. \quad (4.15)$$

Therefore, by Theorem 4.3 along with (4.12) and (4.15), we have, since $I(f_{\mu,\nu}^{(p)}) = \text{SW}_p^p(\mu, \nu)$, the rate

$$\left| \text{SHCV}_{n,L}^p(\mu, \nu) - \text{SW}_p^p(\mu, \nu) \right| = O_{\mathbb{P}}(L^{-1} n^{-1/2}).$$

4.8.4 Additional results on spherical harmonics

Spherical harmonics in small dimensions

Circular harmonics on $\mathbb{S}^1$. On the unit circle, the spherical harmonics are often called *circular harmonics* and correspond to the Fourier basis defined in

circular coordinates $(x, y) = (\cos(t), \sin(t))$ by $\psi_0(\theta) = 1/\sqrt{2\pi}$ and for $k > 0$

$$\psi_{2k}(\theta) = \frac{\sin(kt)}{\sqrt{\pi}}, \quad \psi_{2k-1}(\theta) = \frac{\cos((k-1)t)}{\sqrt{\pi}}.$$

The circular harmonics are $\varphi_{1,0} = \psi_0$ and for $\ell \geq 1$, $\varphi_{\ell,1} = \psi_{2\ell-1}$ and $\varphi_{\ell,2} = \psi_{2\ell}$.

Spherical harmonics on $\mathbb{S}^2$. On the sphere $\mathbb{S}^2$ there exist $N_\ell^3 = 2\ell+1$ linearly independent complex spherical harmonics $\psi_{\ell,k} : \mathbb{S}^2 \rightarrow \mathbb{C}$ of degree ℓ . Using the spherical coordinates $(x, y, z) = (\cos(u) \sin(v), \sin(u) \sin(v), \cos(v))$ with $0 \leq u < 2\pi$ and $0 \leq v < \pi$, the Laplace spherical harmonics are generally defined for all $-\ell \leq k \leq \ell$ by

$$\psi_{\ell,k}(u, v) = \sqrt{\frac{(2\ell+1)(\ell-k)!}{4\pi(\ell+k)!}} P_\ell^k(\cos(v)) e^{iku},$$

where P_ℓ^k is the associated Legendre polynomial defined by $P_\ell^k(x) = (-1)^k (1-x^2)^{k/2} (d^k/dx^k) P_\ell(x)$ with P_ℓ the Legendre polynomial of degree ℓ and $P_\ell^{-k}(x) = (-1)^k [(\ell-k)!/(\ell+k)!] P_\ell^k(x)$. This definition ensures orthonormality $\int_{v=0}^\pi \int_{u=0}^{2\pi} \psi_{\ell,k} \psi_{\ell',k'}^* d\Omega = \delta_{\ell,\ell'} \delta_{k,k'}$ with $d\Omega = \sin(v) du dv$. A real basis of spherical harmonics $\varphi_{\ell,k} : \mathbb{S}^2 \rightarrow \mathbb{R}$ can be defined in terms of their complex analogues by setting $\varphi_{\ell,0} = \psi_{\ell,0}$ and taking the real and imaginary parts as $\varphi_{\ell,k} = \sqrt{2}(-1)^k \Im(\psi_{\ell,|k|})$ if $k < 0$ and $\varphi_{\ell,k} = \sqrt{2}(-1)^k \Re(\psi_{\ell,k})$ if $k > 0$.

Number of spherical harmonics

Figure 4.9a presents the evolution of the number N_ℓ^d of spherical harmonics of degree ℓ in dimension d with $d \in [3, 10]$ and $\ell \in [1, 10]$ while Figures 4.9b and 4.9c report the evolution of the cumulative sum $\sum_{\ell=1}^L N_\ell^d$ and even cumulative sum $\sum_{\ell=1}^L N_{2\ell}^d$ for maximum degree $L \in [1, 10]$ and $2L \in \{2; 4; 6; 8; 10; 12\}$ respectively.

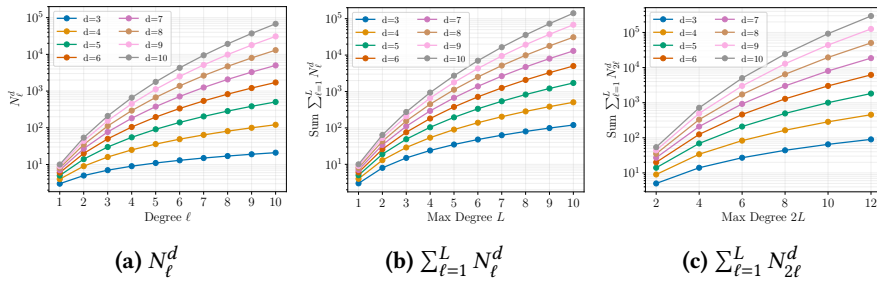


Figure 4.9: Evolution of the number N_ℓ^d of spherical harmonics and (even) cumulative sums according to the degree.

Table 4.9 reports the maximal degree of spherical harmonics and number of control variates for the considered dimensions in the experiments (See Lemma 4.1). For $d \in \{10; 20\}$ we only consider a subset of spherical harmonics of degree 4 to obtain a total number of $s = 300$ control variates.

dimension d	3	5	6	10	20
max degree $2L$	16	6	4	4	4
number s of control variates	152	209	125	714	8854

Table 4.9: Parameter setting of control variates.

Spherical harmonics and Gegenbauer polynomials

For $\alpha > 0$, Gegenbauer polynomials of degree ℓ , denoted by $C_\ell^\alpha : [-1, 1] \rightarrow \mathbb{R}$ are orthogonal polynomials with respect to the weight function $z \mapsto (1 - z^2)^{\alpha-1/2}$. They are special cases of Jacobi polynomials and can be written as

$$C_\ell^\alpha(z) = \sum_{k=0}^{\lfloor \ell/2 \rfloor} \frac{(-1)^k \Gamma(\ell - k + \alpha)}{\Gamma(\alpha) k! (\ell - 2k)!} (2z)^{\ell-2k}, \quad C_\ell^\alpha(1) = \frac{\Gamma(2\alpha + \ell)}{\Gamma(2\alpha)\ell!}. \quad (4.16)$$

In the following we fix ℓ , set $N = N_\ell^d$, and fix $\{\varphi_1, \dots, \varphi_N\}$ as an orthonormal basis for $\mathcal{H}_\ell^d$. Denote by $Z_\ell(\cdot, \cdot)$ the reproducing kernel of $\mathcal{H}_\ell^d$. In terms of orthonormal basis we have

$$\forall \theta, \eta \in \mathbb{S}^{d-1}, \quad Z_\ell(\theta, \eta) = \sum_{k=1}^N \varphi_k(\theta) \varphi_k(\eta).$$

Interestingly, the spherical harmonics are linked to the Gegenbauer polynomials through the following addition formula

Theorem 4.4. (Dai, 2013, Theorem 1.2.6) For $\ell \geq 0$ and $\theta, \eta \in \mathbb{S}^{d-1}$, $d \geq 3$

$$Z_\ell(\theta, \eta) = \sum_{k=1}^N \varphi_k(\theta) \varphi_k(\eta) = \frac{\ell + \alpha}{\alpha} C_\ell^\alpha(\langle \theta, \eta \rangle), \quad \alpha = \frac{d-2}{2}.$$

Corollary 4.3. (Dai, 2013, Corollary 1.2.8) For $\alpha = \frac{d-2}{2}$, the Gegenbauer polynomials C_ℓ^α satisfy the orthogonality relation

$$\frac{|\mathbb{S}^{d-2}|}{|\mathbb{S}^{d-1}|} \int_{-1}^1 C_\ell^\alpha(t) C_{\ell'}^\alpha(t) (1-t^2)^{\alpha-\frac{1}{2}} dt = h_\ell^\alpha \delta_{\ell, \ell'}, \quad h_\ell^\alpha = \frac{\alpha}{\ell + \alpha} C_\ell^\alpha(1).$$

Let $\{\theta_1, \dots, \theta_N\}$ be a collection of points on $\mathbb{S}^{d-1}$. The following remarks motivate the definition of a fundamental set. We let $M_1 := \varphi_1(\theta_1)$ and for $k = 2, 3, \dots, N$, define matrices

$$M_k := \begin{bmatrix} \varphi_1(\theta_1) & \dots & \varphi_1(\theta_k) \\ \vdots & \dots & \vdots \\ \varphi_k(\theta_1) & \dots & \varphi_k(\theta_k) \end{bmatrix},$$

$$M_k(\theta) := \left[\begin{array}{c|c} & \varphi_1(\theta) \\ & \vdots \\ M_{k-1} & \varphi_{k-1}(\theta) \\ \hline \varphi_k(\theta_1), \dots, \varphi_k(\theta_{k-1}) & \varphi_k(\theta) \end{array} \right].$$

The product of M_N and its transpose $M_N^\top$ can be summed, on applying the addition formula, as $M_N^\top M_N = [Z_\ell(\theta_i, \theta_j)]_{i,j=1}^N$, which shows, in particular, $\det [Z_\ell(\theta_i, \theta_j)]_{i,j=1}^N = (\det M_N)^2 \geq 0$.

Definition 4.5. (Dai, 2013, Definition 1.3.1) A collection of points $\{\theta_1, \dots, \theta_N\}$ in $\mathbb{S}^{d-1}$ is called a fundamental system of degree ℓ on the sphere $\mathbb{S}^{d-1}$ if

$$\det [C_\ell^\alpha(\langle \theta_i, \theta_j \rangle)]_{i,j=1}^N > 0, \quad \alpha = \frac{d-2}{2}$$

Theorem 4.5. (Dai, 2013, Theorem 1.3.3) If $\{\theta_1, \dots, \theta_N\}$ is a fundamental system of points on the sphere, then $\{C_\ell^\alpha(\langle \cdot, \theta_i \rangle) | i = 1, \dots, N\}$, $\alpha = (d-2)/2$ is a basis of $\mathcal{H}_\ell^d|_{\mathbb{S}^{d-1}}$

This theorem lies at the heart of the practical implementation used in Dutordoir et al. (2020) which constructs a fundamental system of vectors for efficient evaluation of spherical harmonics. The mathematical aspects of this process involve concepts such as the Gegenbauer polynomial, Cholesky decomposition, optimization, and normalization. The goal is to construct a set of vectors efficiently spanning the space required for evaluating spherical harmonics in the given dimension and degree. The procedure works as follows. Start by defining the Gegenbauer polynomial, denoted as $C_\ell^\alpha(\theta)$, of a specific degree ℓ and a dimension-dependent parameter $\alpha = (d-2)/2$. Select the first vector arbitrarily, often chosen as the north pole or the unit vector along the last dimension. Compute the Cholesky decomposition of the Gegenbauer-gram matrix associated with the current vectors and iteratively add new vectors to to expand the fundamental system. For each new vector, optimize the determinant of the Gegenbauer-gram matrix, measuring the span of the space generated by the vectors. The optimization process involves finding a new vector that maximizes the determinant of the augmented matrix. This is typically done using optimization techniques such as BFGS. After finding the optimal vector, normalize it to ensure it has unit length.

Simple case study for discrete measures and Gaussian measures

To investigate the approximation power of the integrand $\theta \mapsto f_{\mu,\nu}^{(p)}(\theta) = W_p^p(\theta_{\#}^{\star}\mu_2, \theta_{\#}^{\star}\nu_2)$ via spherical harmonics, we consider a case study in dimension $d = 2$ on the unit circle $\mathbb{S}^1$ with circular harmonics $(\varphi_{\ell,k})$ (see Section 4.8.4). Using an Ordinary Least Squares regression, the polynomial approximation $\widehat{f}_{2L}$ of $f_{\mu,\nu}^{(p)}$ can be written as

$$\widehat{f}_{2L}(\theta) = \alpha + \sum_{\ell=2,4,\dots,2L} \left(\beta_{\ell}^{(1)} \varphi_{\ell,1}(\theta) + \beta_{\ell}^{(2)} \varphi_{\ell,2}(\theta) \right)$$

where α and $\beta = (\beta_2^{(1)}, \beta_2^{(2)}, \dots)$ are estimated by the OLS procedure. We investigate the smoothness properties of the integrand $f_{\mu,\nu}^{(p)}$ by comparing different measures μ and ν . We first consider the case of discrete measures μ_2 and ν_2 each supported on two atoms. In this simple case, the integrand $f_{\mu,\nu}^{(p)}(\theta)$ is only Lipschitz and we study the approximation error of our method. Then we consider the case of Gaussian measures $\mu = \mathcal{N}_2(\mathbf{m}_X, \Sigma_X)$ and $\nu = \mathcal{N}_2(\mathbf{m}_Y, \Sigma_Y)$. In that case, the integrand of interest is infinitely smooth.

Discrete measures. We consider the discrete measures μ_2 and ν_2 associated to the points $x_1 = (0, 0)$, $x_2 = (1, 0)$ and $y_1 = (0, 0)$, $y_2 = (1, 1)$ respectively. We have $\mu_2 = \frac{1}{2}\delta_{(0,0)} + \frac{1}{2}\delta_{(1,0)}$ and $\nu_2 = \frac{1}{2}\delta_{(0,0)} + \frac{1}{2}\delta_{(1,1)}$. The integrand $f_{\mu,\nu}^{(p)}$ can be computed exactly as

$$W_p^p(\theta_{\#}^{\star}\mu_2, \theta_{\#}^{\star}\nu_2) = \begin{cases} \frac{1}{2} |\sin(\theta)|^p, & \theta \in D_1, \\ \frac{1}{2} (|\cos(\theta)|^p + |\cos(\theta) + \sin(\theta)|^p), & \theta \in D_2, \end{cases}$$

where $D_1 = [0, \pi/2] \cup [3\pi/4, 3\pi/2] \cup [7\pi/4, 2\pi]$ and $D_2 = [\pi/2, 3\pi/4] \cup [3\pi/2, 7\pi/4]$. By focusing on the case of order $p = 2$, we have

$$\begin{aligned} I_1 &= \int_0^{\pi/2} \sin^2(\theta) d\theta = \frac{\pi}{4}, \\ I_2 &= \int_{\pi/2}^{3\pi/4} \cos^2(\theta) + (\cos(\theta) + \sin(\theta))^2 d\theta = \frac{3\pi}{8} - \frac{3}{4}, \\ I_3 &= \int_{3\pi/4}^{3\pi/2} \sin^2(\theta) d\theta = \frac{3\pi}{8} - \frac{1}{4}, \\ I_4 &= \int_{3\pi/2}^{7\pi/4} \cos^2(\theta) + (\cos(\theta) + \sin(\theta))^2 d\theta = \frac{3\pi}{8} - \frac{3}{4}, \\ I_5 &= \int_{7\pi/4}^{2\pi} \sin^2(\theta) d\theta = \frac{\pi}{8} - \frac{1}{4}. \end{aligned}$$

Therefore,

$$\begin{aligned} \text{SW}_2^2(\mu_2, \nu_2) &= \frac{1}{2\pi} \int_0^{2\pi} W_2^2(\theta_\#^* \mu_2, \theta_\#^* \nu_2) d\theta \\ &= \frac{1}{4\pi} (I_1 + I_2 + I_3 + I_4 + I_5) \\ &= \frac{3}{8} - \frac{1}{2\pi}. \end{aligned}$$

Recall that $\text{SHCV}_{n,L}^p(\mu_2, \nu_2) = I_n^{\text{ols}}$ where $(I_n^{\text{ols}}, \beta_n) \in \arg \min_{\alpha, \beta} \|f_n - \alpha \mathbf{1}_n - \Phi \beta\|_2^2$.

Figure 4.10a reports the evolution of $W_2^2(\theta_\#^* \mu_2, \theta_\#^* \nu_2)$ (black line) and its polynomial approximation $\widehat{f}_{2L}(\theta)$ with different degrees of harmonics $2L \in \{2; 4; 8; 10\}$. Figure 4.10b displays the approximation error, i.e., the difference ε_n between $W_2^2(\theta_\#^* \mu_2, \theta_\#^* \nu_2)$ and its polynomial approximation $\widehat{f}_{2L}(\theta)$.

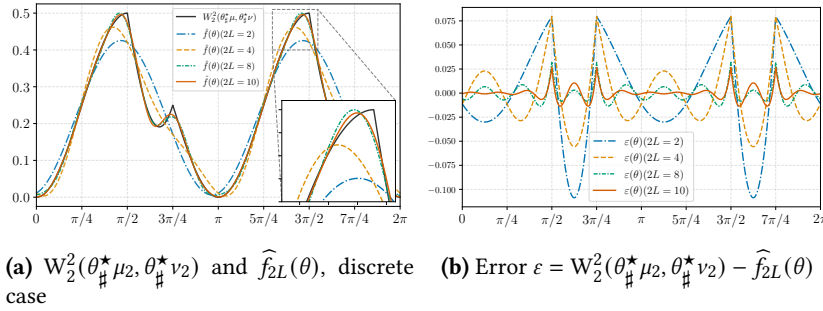


Figure 4.10: $W_2^2(\theta_\#^* \mu_2, \theta_\#^* \nu_2)$, polynomial approximation $\widehat{f}_{2L}(\theta)$ and errors $\varepsilon = W_2^2(\theta_\#^* \mu_2, \theta_\#^* \nu_2) - \widehat{f}_{2L}(\theta)$ for two discrete measures.

Gaussian measures. Let $\mu = \mathcal{N}_d(\mathbf{m}_X, \Sigma_X)$ and $\nu = \mathcal{N}_d(\mathbf{m}_Y, \Sigma_Y)$. Then, for $\theta_\#^* \mu = \mathcal{N}_1(\theta^\top \mathbf{m}_X, \theta^\top \Sigma_X \theta)$ and $\theta_\#^* \nu = \mathcal{N}_1(\theta^\top \mathbf{m}_Y, \theta^\top \Sigma_Y \theta)$, we have $W_2^2(\theta_\#^* \mu, \theta_\#^* \nu) = |\theta^\top (\mathbf{m}_X - \mathbf{m}_Y)|^2 + (\sqrt{\theta^\top \Sigma_X \theta} - \sqrt{\theta^\top \Sigma_Y \theta})^2$. We consider $\mathbf{m}_X = (0, 0)^\top$, $\mathbf{m}_Y = (1, 1)^\top$, $\Sigma_X = \begin{bmatrix} 1 & 0.2 \\ 0.2 & 1 \end{bmatrix}$, $\Sigma_Y = \begin{bmatrix} 10 & 3 \\ 3 & 10 \end{bmatrix}$

Figure 4.11a reports the evolution of $W_2^2(\theta_\#^* \mu_2, \theta_\#^* \nu_2)$ (black line) and its polynomial approximation $\widehat{f}_{2L}(\theta)$ with different degrees of harmonics $2L \in \{2; 4; 6\}$. Observe that in this case the integrand is very smooth so that the polynomial approximations seem to overlap. Figure 4.11b displays the approximation error, i.e., the difference ε_n between $W_2^2(\theta_\#^* \mu_2, \theta_\#^* \nu_2)$ and its polynomial approximation $\widehat{f}_{2L}(\theta)$.

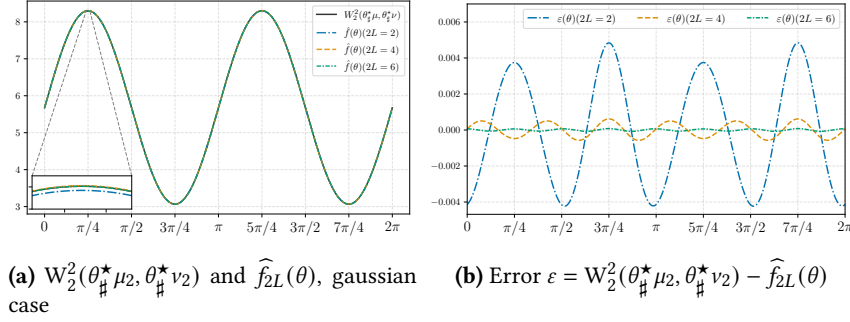


Figure 4.11: $W_2^2(\theta_\#^* \mu_2, \theta_\#^* v_2)$, polynomial approximation $\widehat{f}_{2L}(\theta)$ and errors $\varepsilon = W_2^2(\theta_\#^* \mu_2, \theta_\#^* v_2) - \widehat{f}_{2L}(\theta)$ for two Gaussian measures.

Figure 4.12a presents the boxplots of the error distribution ($\text{SHCV}_{n,L}^2 - \text{SW}_2^2$) both for discrete measures μ_2, v_2 and for Gaussian measures with different degrees of harmonics $2L \in \{2; 4; 6; 8\}$ and Monte Carlo sample size $n = 10^4$. The Sliced-Wasserstein distance for Gaussian measures is obtained by the naive Monte Carlo method with $n = 10^8$. The boxplots are obtained over 100 replications.

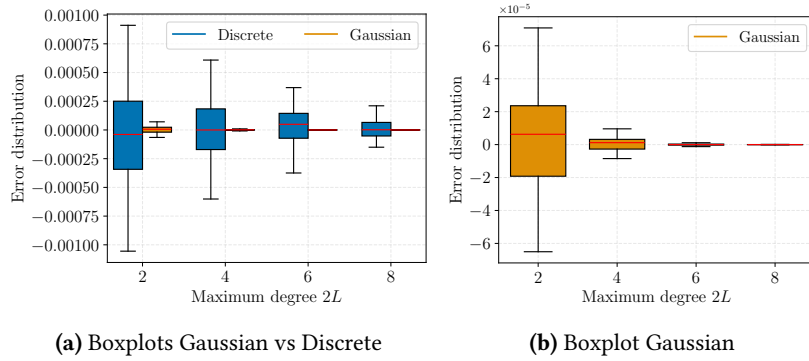


Figure 4.12: Error distribution of $(\text{SHCV}_{n,L}^2 - \text{SW}_2^2)$.

Numerical check for exact integration rule

We compute $\text{SW}_2^2(\mu, \nu)$ for $\mu = \mathcal{N}(a, \mathbf{A})$ and $\nu = \mathcal{N}(b, \mathbf{B})$ with means $a, b \sim \mathcal{N}(1_d, I_d)$ and proportional (see Section 4.6.1) covariance matrices $\mathbf{A} = \Sigma_a \Sigma_a^\top$ and $\mathbf{B} = \gamma \mathbf{A}$ where all the entries of Σ_a are drawn according to $\mathcal{N}(0, 1)$. We consider dimensions $d \in \{3; 5\}$ and a number of projections $n \in [10^2; 10^4]$. Observe that in this case, the integrand of interest is a quadratic polynomial

in θ given by

$$\begin{aligned} f_{\mu, \nu}^{(2)}(\theta) &= |\theta^\top (a - b)|^2 + (\sqrt{\theta^\top \mathbf{A} \theta} - \sqrt{\theta^\top \mathbf{B} \theta})^2 \\ &= |\theta^\top (a - b)|^2 + (1 - \sqrt{\gamma})^2 (\theta^\top \mathbf{A} \theta), \end{aligned}$$

and $\text{SW}_2^2(\mu, \nu)$ is given in close-form by integrating over $\theta \in \mathbb{S}^{d-1}$ such that

$$\text{SW}_2^2(\mu, \nu) = \frac{\|a - b\|_2^2}{d} + \frac{(1 - \sqrt{\gamma})^2}{d} \text{Tr}(\mathbf{A}).$$

Figure 4.13 reports the mean squared errors for the different estimates $\widehat{\text{SW}}_n(\mu, \nu)$ where the expectation is computed as an average over 100 independent runs with parameter $\gamma = 2$. Observe that, while the standard Monte Carlo estimate and other control variate-based methods achieve a small MSE in the range $[10^{-7}; 10^{-3}]$, our SHCV estimate obtains a MSE of 10^{-30} which is virtually zero up to machine precision. This numerical comparison empirically validates the exact integration rule of Proposition 4.2.

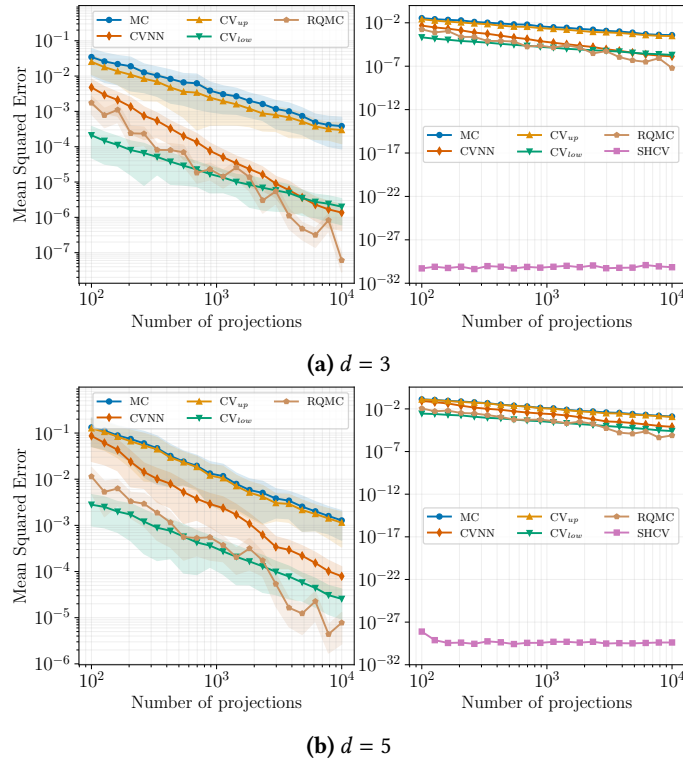


Figure 4.13: MSE for Gaussian distributions with proportional dispersion matrices, dimension $d \in \{3; 5\}$, obtained over 100 replications.

Random Forest Control Variates

| 5

A random forest is an ensemble of trees where each tree contributes to the collective decision-making process. Due to the nature of decision trees in a random forest, the decision boundaries are typically rectangular, with each rectangle having a constant prediction. This property makes random forests particularly suitable for constructing control variates, which are special functions with known expectations used for variance reduction in Monte Carlo integration. However, various types of random forests exist. In this work, we specifically consider three classical types: Random Forests, Bagging, and Extremely Randomized Trees. We derive three working estimates and conduct simulation studies, demonstrating that our proposed estimates outperform the classical Monte Carlo estimate.

5.1 Introduction

Consider the classical integration problem $\mu(f) = \int_{\mathbb{R}^d} f(x) \mu(dx)$, where μ is a probability measure on $\mathbb{R}^d$ and $f \in L_2(\mu)$. The standard Monte Carlo estimate approximates this value by using an independent random sample $X_1, \dots, X_n$ drawn from μ and takes the average

$$\hat{\mu}_n^{(\text{MC})}(f) = \frac{1}{n} \sum_{i=1}^n f(X_i).$$

The rate of convergence in probability of the standard Monte Carlo estimate is $O(n^{-1/2})$.

To reduce the variance of the estimate, we apply the method of control variates (Glasserman, 2004; Caflisch, 1998). This method consists of two steps:

1. Build a surrogate function $\hat{f}$ with a known integral $\mu(\hat{f})$.
2. Use the centered variables $\hat{f}(X_i) - \mu(\hat{f})$ to build the control variate Monte Carlo estimate

$$\hat{\mu}_n^{(\text{CV})}(f) = \frac{1}{n} \sum_{i=1}^n \left[f(X_i) - \left\{ \hat{f}(X_i) - \mu(\hat{f}) \right\} \right].$$

We propose to construct such a surrogate function by means of a random forest, that is, an ensemble of decision trees. As we deal with a continuous response $y_i = f(X_i)$, $i = 1, \dots, n$, we will opt for regression trees (Breiman et al., 1984) to estimate $\hat{f}$. The idea behind this approach is to recursively partition the feature space into regions, where each region is associated with a constant value prediction for the target variable. This involves splitting the data at each node based on a chosen criterion, typically aiming to minimize the mean squared error of the target variable within each split.

The approach shares similarities with k-nearest neighbors (Biau and Devroye, 2015), as both methods partition the input space to predict outcomes based on local data subsets. In Chapter 3, we construct the surrogate function using the nearest neighbor estimate. However, integrating such estimator requires computing Voronoi volumes, which can be computationally intensive. In contrast, our new approach addresses this challenge by using rectangular partitions. Calculating the volume of a rectangle offers a more computationally efficient solution.

The motivation for using an ensemble of trees instead of a single regression tree is to improve accuracy by combining the outputs of multiple trees. The collective decision-making process of many trees leads to more reliable and consistent predictions. Among the ensembles of trees, we consider the following: Random Forests (Breiman, 2001), Bagging (Breiman, 1996), and Extremely Randomized Trees (Geurts et al., 2006). The key feature of Random Forests and Bagging is that they use bootstrap replications of the training sample to build individual trees. However, Random Forests also employ feature selection at each split. Extremely Randomized Trees, on the contrary, use the entire learning sample but add randomization in the split procedure.

The theoretical guarantees of Random Forests remain an open question. However, the rate of convergence for a simplified model called Centered Random Forests (Breiman, 2004) has been discussed in Biau (2012) and Klusowski (2021). Centered Forests operate under the following principles: (i) the input space is $[0, 1]^d$; (ii) it does not involve a resampling step; (iii) at each node of every individual tree, a coordinate is uniformly chosen from $\{1, \dots, d\}$; and (iv) a split is made at the center of the cell along the selected coordinate.

In this work, we aim to understand how tuning parameters influence the accuracy of the proposed estimates. Specifically, we consider the number of trees and the maximum depth of each tree as tuning parameters. To this end, we conduct experiments on two simple integration problems: integration with respect to the uniform distribution on a unit cube and integration with respect to the normal distribution. Additionally, we consider integration with respect to a mixture of two Gaussian distributions. Furthermore, we apply the proposed estimates to calculate the variance of the asymptotic distribution of

U-statistics.

The outline of the chapter is as follows. Section 5.2 describes the decision tree and random forest methods. In Section 5.3, we explain the mechanism of control variate construction and identify some applications. In Section 5.4, we perform the numerical studies. Additional numerical results are provided in the Appendix.

5.2 Decision trees and random forests

Before we dive into decision trees, let us begin by introducing some basic terms.

Definition 5.1 (Root node). *The root node is the initial node at the beginning of a decision tree, where the entire population or dataset starts dividing based on various features or conditions.*

Definition 5.2 (Decision nodes). *Nodes resulting from the splitting of the root node are known as decision nodes. These nodes represent intermediate decisions or conditions within the tree.*

Definition 5.3 (Leaf nodes). *Leaf nodes are the nodes where further splitting is not possible, often indicating the final classification or outcome. Leaf nodes are also referred to as terminal nodes.*

Definition 5.4 (Parent and child nodes). *In a decision tree, a node that is divided into sub-nodes is known as a parent node, and the sub-nodes emerging from it are referred to as child nodes. The parent node represents a decision or condition, while the child nodes represent the potential outcomes or further decisions based on that condition.*

5.2.1 Regression trees

A decision tree is a model that makes predictions based on a series of binary splits in the feature space. Here are the key principles for constructing a decision tree:

1. **Recursive binary splitting:** The process begins with the entire dataset. The algorithm recursively splits the data at each node into two child nodes. The split is based on the feature that results in the greatest reduction in a certain criterion, typically the mean squared error (MSE) for regression tasks.
2. **Selection of splits:** For each feature, the algorithm evaluates all possible split points. For each candidate split, it calculates the reduction in MSE

from the parent node to the child nodes. The split that maximizes this reduction in MSE is chosen. See Algorithm 5.1.

3. **Stopping criteria:** The recursive splitting continues until one of the stopping criteria is met. Common stopping criteria include:
 - A maximum depth of the tree is reached.
 - The number of samples in a node is less than a minimum threshold.
4. **Prediction:** For regression trees, the prediction for a new observation is the mean response of the training samples in the leaf node that the observation falls into.

Algorithm 5.1 Finding the best split in a node

Require: Features X_{ij} , $i = 1, \dots, n$, $j = 1, \dots, d$, responses y_i , $i = 1, \dots, n$

- 1: Initialize variables `best_feature`, `best_threshold`, `best_mse`
- 2: **for** $j = 1$ to d **do**
- 3: Sort the feature values X_{ij} among all points i in the current node
- 4: **for** each unique value t in the sorted feature values **do**
- 5: Split the dataset into two subsets: $S_L = \{(X_i, y_i) \mid X_{ij} \leq t\}$ and $S_R = \{(X_i, y_i) \mid X_{ij} > t\}$
- 6: Calculate the MSE for this split as $\text{MSE}(S) = \frac{1}{|S|} \left(\sum_{i \in S_L} (y_i - \bar{y}_{S_L})^2 + \sum_{i \in S_R} (y_i - \bar{y}_{S_R})^2 \right)$
- 7: **if** $\text{MSE}(S) < \text{best_mse}$ **then**
- 8: Update `best_feature`, `best_threshold`, `best_mse`
- 9: **end if**
- 10: **end for**
- 11: **end for**
- 12: **return** `best_feature`, `best_threshold`, `best_mse`

From a mathematical perspective, the given set of possible values $X_1, \dots, X_n$ are divided into J distinct and non-overlapping regions $R_1, \dots, R_J$. For a given observation falling into the region R_j , the prediction is equal to the mean of the response values y_i for all n_j training observations in the region R_j , i.e.,

$$\hat{y}_j = \frac{1}{n_j} \sum_{X_i \in R_j} y_i.$$

Therefore, $\hat{f}(X_i) = \hat{y}_j$ for all $X_i \in R_j$. Figure 5.1 shows the representation of a regression tree and its boundaries where F_1, F_2 are features, s_1, s_2, s_3 are thresholds and R_1, R_2, R_3, R_4 are decision regions.

In order to calculate the integral $\mu(\hat{f})$, we need to obtain the boundaries of the regions $R_1, \dots, R_J$. We know that $\hat{f}$ is constant on each of these regions, $\hat{f}(x) = \sum_{j=1}^J \hat{y}_j \mathbb{1}_{R_j}(x)$. Therefore,

$$\mu(\hat{f}) = \sum_{j=1}^J \hat{y}_j \mu(R_j),$$

where $\mu(R_j) = \int_{R_j} \mu(dx)$ is the volume of R_j . For retrieving the boundaries of each R_j , we start at the leaf nodes and move upwards. When the child node is

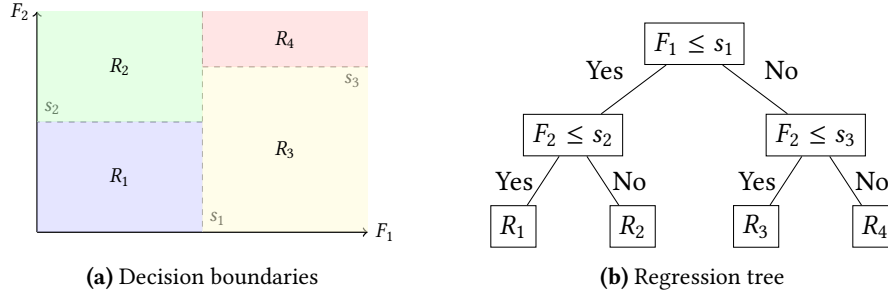


Figure 5.1: Regression tree and its decision boundaries

the one on the left (a condition of the form $F_k \leq s_k$ is satisfied), we establish the feature boundary on the right. Similarly, if the child node is the one on the right (a condition of the form $s_k < F_k$ is satisfied), we assign the boundary on the left. This procedure takes into account the cases where the feature may have been previously selected for splitting, and a boundary has already been assigned. Since R_j is a rectangle, $R_j = \prod_{k=1}^d \{F_k : a_{kj} < F_k \leq b_{kj}\}$ with thresholds $-\infty \leq a_{kj} < b_{kj} \leq \infty$, calculating the volume $\mu(R_j)$ is easiest when μ is a product measure, $\mu = \otimes_{k=1}^d \mu_k$: then

$$\mu(R_j) = \prod_{k=1}^d \mu_k(\{F_k : a_{kj} < F_k \leq b_{kj}\}). \quad (5.1)$$

5.2.2 Random forests

The lower-case term *random forests* is employed to describe any ensemble of decision trees, whereas the ensemble method introduced by Breiman (2001) is specifically referred to as *Random Forests* with capital letters. Here, we present some types of random forests:

- **Bagging (Bootstrap Aggregating)** (Breiman, 1996): Bagging involves training decision trees on bootstrap samples of the training set. Bootstrap sampling randomly selects observations with replacement, creating multiple datasets of the same size but with potential repetition of certain data points. Each decision tree in the Bagging ensemble is trained on one of these bootstrap samples. By combining predictions from multiple trees, Bagging aims to improve model stability and accuracy. This is achieved by reducing the variance of the base learner, which is often high when using models like decision trees that can overfit the training data.
- **Random Forests** (Breiman, 2001): Random Forests extend Bagging by introducing additional randomness in the feature selection process during tree construction. Instead of considering all features for determining

the best split at each node, Random Forests randomly select a subset of K features from the total d features. This random feature subset selection promotes diversity among the trees, thereby reducing correlation between them and improving generalization ability. While the standard Algorithm 5.1 considers all features to find the best split, in Random Forests, K features are randomly selected after initialization and used for subsequent steps, enhancing the model's robustness and performance across various datasets and learning tasks.

- **Extremely Randomized Trees** (Geurts et al., 2006): Extremely Randomized Trees (or Extra-Trees) differ from Random Forests in that they do not use bootstrap samples. Instead, the entire training set is used to build each tree. The algorithm selects random thresholds for each feature, chosen uniformly at random between the maximum and minimum values of that feature. Using these randomly selected thresholds, the algorithm then searches for the best split among them (see Algorithm 5.2). This approach reduces computational complexity and can lead to faster training times. It also includes random feature selection. In the special case where $K = 1$, *Extremely Randomized Trees* reduce to *Totally Randomized Trees*, in which both a feature and a threshold are drawn at random at each node.

Algorithm 5.2 Finding the best random split in a node

Require: Features X_{ij} , $i = 1, \dots, n$, $j = 1, \dots, d$, responses y_i , $i = 1, \dots, n$

```

1: Initialize variables best_feature, best_threshold, best_mse
2: for  $j = 1$  to  $d$  do
3:   Determine the range of the feature values  $X_{ij}$ :  $[\min(X_{ij}), \max(X_{ij})]$  among all points  $i$  in the
   current node
4:   Generate  $K$  random thresholds  $t_k$  uniformly within this range
5:   for each random threshold  $t_k$  do
6:     Split the dataset into two subsets:  $S_L = \{(X_i, y_i) \mid X_{ij} \leq t_k\}$  and  $S_R = \{(X_i, y_i) \mid X_{ij} > t_k\}$ 
7:     Calculate the MSE for this split as  $\text{MSE}(S) = \frac{1}{|S|} \left( \sum_{i \in S_L} (y_i - \bar{y}_{S_L})^2 + \sum_{i \in S_R} (y_i - \bar{y}_{S_R})^2 \right)$ 
8:     if  $\text{MSE}(S) < \text{best\_mse}$  then
9:       Update best_feature, best_threshold, best_mse
10:    end if
11:  end for
12: end for
13: return best_feature, best_threshold, best_mse

```

The time complexity for building a single tree and forests of M randomized trees is stated in Table 5.1 for two cases (Louppe, 2014):

- *Best case.* In this situation, we consider the most efficient procedure when node samples can always be partitioned into two balanced subsets of $N/2$ observations where N is the number of observations in the current node.
- *Worst case.* This case is characterized by the least efficient procedure when splits are unbalanced. In the worst-case scenario, this results in splits that move a single observation in the first sub-tree and put all

$N - 1$ others in the second sub-tree.

As Bagging and Random Forests are constructed on bootstrap samples, we have $\tilde{n} = 0.632n$ in Table 5.1 due to the fact that bootstrap samples draw, on average, 63.2% of unique samples.

	Best case	Worst case
Single tree	$O(dn \log^2(n))$	$O(dn^2 \log(n))$
Bagging	$O(Md\tilde{n} \log^2(\tilde{n}))$	$O(Md\tilde{n}^2 \log(n))$
Random Forests	$O(MK\tilde{n} \log^2(\tilde{n}))$	$O(MK\tilde{n}^2 \log(\tilde{n}))$
Extremely Randomized Trees	$O(MKn \log(n))$	$O(MKn^2)$

Table 5.1: Time complexity for building forests.

5.3 Control variates

The idea is to consider various regression trees to obtain optimal control variates. For instance, one can explore both fully grown trees and trees with limited maximum depth. The reason for considering the latter ones is to reduce memory consumption, the complexity and size of the tree. Note that if we use the full sample $X_1, \dots, X_n$ as the learning sample to construct a single tree or Extremely Randomized Trees we obtain $\sum_{i=1}^n f(X_i) = \sum_{i=1}^n \hat{f}(X_i)$. This comes from the fact that, for each region R_j ,

$$\sum_{X_i \in R_j} \hat{f}(X_i) = \sum_{X_i \in R_j} \frac{1}{n_j} \sum_{X_i \in R_j} y_i = \sum_{X_i \in R_j} f(X_i).$$

Therefore, we have

$$\hat{\mu}_n^{(\text{CV})}(f) = \mu(\hat{f}).$$

Regarding the integral $\mu(\hat{f})$, we can calculate it exactly by retrieving the decision boundaries provided μ is a product measure as defined in Eq. (5.1).

Next, we repeat this procedure several times, constructing a new regression tree each time. Eventually, we obtain control variates by aggregating the outputs from each tree, leading to the random forest control variate estimate.

Consider the following examples of applying the proposed approach:

- **Mixture models.** This approach remains valid even when integrating with respect to a mixture probability measure $\mu = \sum_i w_i \mu_i$, where $\sum_i w_i = 1$ and $w_i > 0$, and where each μ_i is a product measure. This is due to the linearity of integration with respect to measures, which allows us to express $\mu(f)$ as $\sum_i w_i \mu_i(f)$.

- **Asymptotic distribution of U-statistics.** Another context where this type of integral arises is when considering a U-statistic, which is represented as

$$U = \frac{1}{\binom{n}{m}} \sum_{1 \leq i_1 < i_2 < \dots < i_m \leq n} h(X_{i_1}, X_{i_2}, \dots, X_{i_m}),$$

where $X_1, \dots, X_n$ is an i.i.d. sample from μ , $m \leq n$ is the order of the U-statistic, and h is a symmetric kernel function. To calculate the variance of the asymptotic distribution of a U-statistic, which is equal to $m^2 \zeta_1$, we need to calculate the following covariance (van der Vaart, 1998):

$$\begin{aligned} \zeta_1 &= \text{cov}(h(X_1, X_2, \dots, X_m), h(X_1, X'_2, \dots, X'_m)) \\ &= E[h(X_1, X_2, \dots, X_m)h(X_1, X'_2, \dots, X'_m)] - \theta^2, \end{aligned}$$

where $X_1, X_2, \dots, X_m, X'_2, \dots, X'_m$ are i.i.d. variables from μ , $\theta = E[h(X_1, X_2, \dots, X_m)]$ and $E[h(X_1, X_2, \dots, X_m)h(X_1, X'_2, \dots, X'_m)]$ is the $(2m - 1)$ -dimensional integral of interest.

5.4 Numerical study

In this section we investigate how different tuning parameters affect the root mean square errors of the proposed estimates. We examine two scenarios of the integration problem $\mu(f) = \int f d\mu$:

- **Integration on $[0, 1]^d$.** Consider the case where the measure μ is the uniform distribution over the unit cube $[0, 1]^d$ and the integrand f is

$$f_1(x_1, \dots, x_d) = \prod_{i=1}^d \log(2)2^{1-x_i} - 1.$$

- **Integration on $\mathbb{R}^d$.** Now consider the integration problem $\int f_2(x)\psi(x) dx$ where $\psi(\cdot)$ denotes the probability density function of the multivariate Gaussian distribution $\mathcal{N}(0, I_d)$ and

$$f_2(x_1, \dots, x_d) = \sin\left(\frac{\pi}{d} \sum_{i=1}^d x_i\right).$$

Next, we consider the following estimates: standard Monte Carlo (MC), Random Forest Control Variates (RFCV), Extremely Randomized Trees Control Variates (ERTCV), and Bagging Control Variates (BCV).

Maximum depth. We first explore the influence of the limited maximum tree depth $D_{\max}$ on the RMSE of each estimate. For the sake of simplicity, we operate with only one tree. We fix the Monte Carlo sample size $n = 10000$

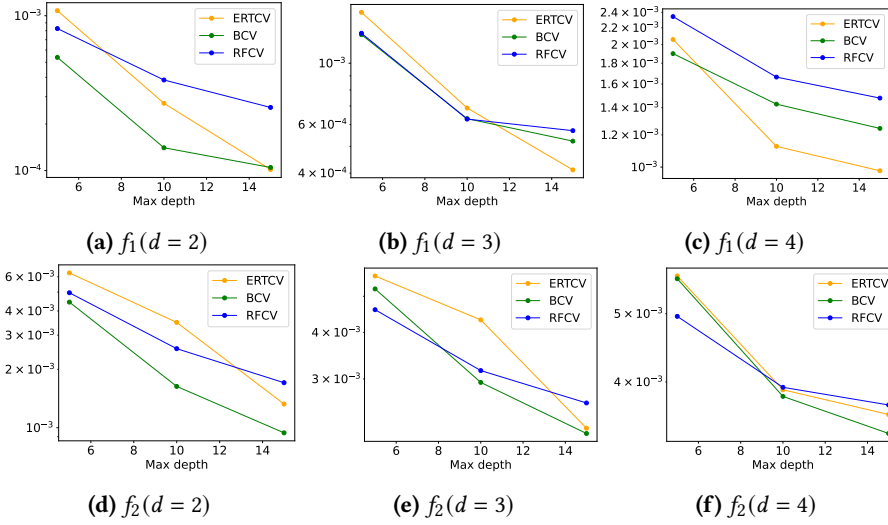


Figure 5.2: Plots of the root mean square errors with different maximum depth obtained over 100 replications

and the dimension of the integration domain $d = 2, 3, 4$. For Random Forests, we set $K = \lceil d/2 \rceil$. The RMSE plots for both scenarios are shown in Figure 5.2.

For the function f_1 , we observe that as the tree depth increases, the RMSE decreases for all three estimates. For shallower depths $D_{\max} = 5, 10$, it is not clear which estimate shows the better performance. However, for deeper trees with $D_{\max} = 15$, ERTCV outperforms the other estimates.

For the function f_2 , we reach the same conclusion about the RMSE as for the uniform scenario. For trees with $D_{\max} = 10, 15$, BCV outperforms the other estimates.

Number of trees. In this case we consider whether using a larger number of trees improves the RMSE of each estimate. We fix $d = 4$ and $n = 1000$. The maximum depth is taken as $\lceil \log(n) \rceil$. In Figure 5.3 you can see the results for both scenarios.

Based on the results, we can conclude that using more than one tree is beneficial. However, increasing the number of trees does not always result in improvement.

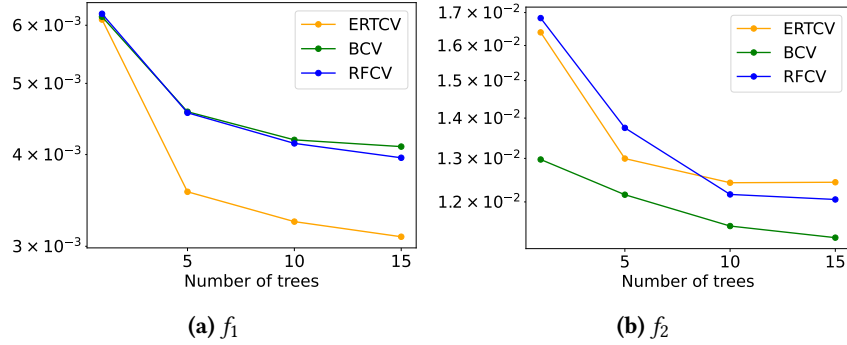


Figure 5.3: Plots of the root mean square errors for ERTCV, BCV and RFCV with different tree numbers obtained over 100 replications

Optimal trees. Taking into account the conclusions concerning the maximum depth and the number of trees, we now construct optimal trees. We perform simulation studies for both scenarios with $d = 2, 3, 4$ and $n = 100, 1000, 10000$. For the control variates, we build fully grown trees since increasing the depth of the trees decreases the RMSE. The number of trees is chosen as $M = 5$. In Figure 5.4, the plots of the root mean square errors for integrands f_1 and f_2 are presented.

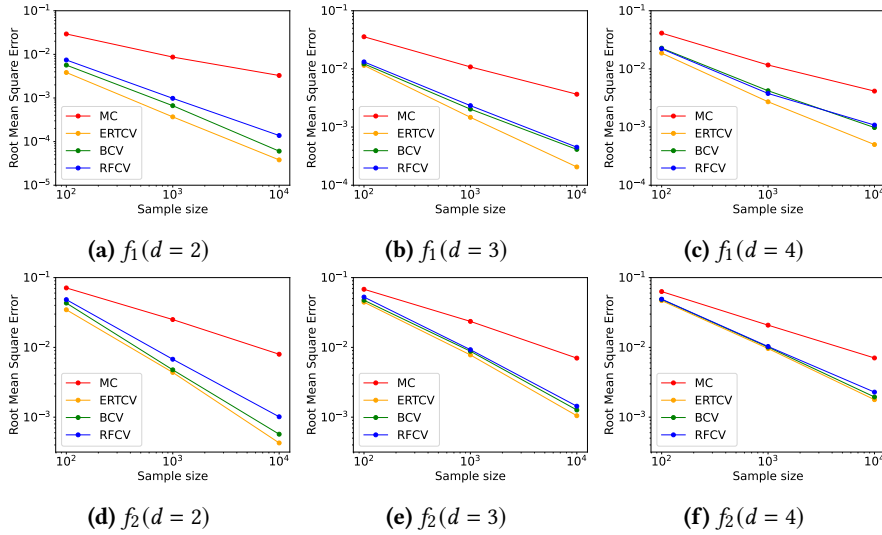


Figure 5.4: Plots of the root-mean square errors obtained over 100 replications with fully grown trees

From Figure 5.4, we can see that all three estimates behave similarly, with ERTCV showing a slight outperformance.

Gaussian mixture model. Consider a mixture of two Gaussian distributions, $0.5\mathcal{N}(-\theta, \Sigma) + 0.5\mathcal{N}(\theta, \Sigma)$, with $\theta = (1, \dots, 1)^\top / (2\sqrt{d})$ and $\Sigma = I_d/d$. The integrands are defined as $f_3 = \sum_{i=1}^d x_i$ and $f_4 = \sum_{i=1}^d x_i^2$. The boxplots of $\hat{\mu}(f_3)$ and $\hat{\mu}(f_4)/\mu(f_4)$ with $n = 1000$ and $d = 2, 3, 4$ are presented in Figures 5.5a and 5.5b respectively. The control variates are built with $M = 5$ fully grown trees.

Signed rank statistic. Consider the U-statistic with the kernel function $h(x_1, x_2) = \mathbf{1}\{x_1 + x_2 > 0\}$ of degree $m = 2$. It can be used as a test statistic for investigating whether the distribution of the observations is located at zero. Under the null hypothesis that the distribution is symmetric about 0, we have $\theta = P(X_1 + X_2 > 0) = 1/2$ and $P(X_1 + X_2 > 0, X_1 + X'_2 > 0) = P(X_1 > -X_2, X_1 > -X'_2) = P(X_1 > X_2, X_1 > X'_2) = 1/3$ since this is just the probability that of three i.i.d. random variables, the first is the largest. Therefore,

$$m^2\zeta_1 = m^2(E[h(X_1, X_2)h(X_1, X'_2)] - \theta^2) = 4(1/3 - (1/2)^2) = 1/3.$$

Consider the case when $X_1, X_2, X'_2 \sim \mathcal{N}(0, 1)$. The boxplots of the ratio $\hat{\zeta}_1/\zeta_1$ are presented in Figure 5.6a. The control variates are built with $M = 5$ fully grown trees.

Kendall's τ . In this example the observations are a sample of bivariate vectors, written as $(X_1, Y_1), \dots, (X_n, Y_n)$. For convenience, we denote them below as $Z_1, \dots, Z_n$. Kendall's τ statistic is

$$\tau = \frac{4}{n(n-1)} \sum_{i < j} \mathbf{1}\{(Y_j - Y_i)(X_j - X_i) > 0\} - 1.$$

This statistic is a measure of dependence between X and Y . The statistic $\tau + 1$ is a U-statistic of order 2 for the kernel

$$h(z_1, z_2) = h\left(\begin{pmatrix} x_1 \\ y_1 \end{pmatrix}, \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}\right) = 2 \cdot \mathbf{1}\{(y_2 - y_1)(x_2 - x_1) > 0\}.$$

If X and Y are independent and have continuous marginal distribution functions, then $E\tau = 0$ and the asymptotic variance

$$4\zeta_1 = 4(E[h(Z_1, Z_2)h(Z_1, Z'_2)] - (\tau + 1)^2)$$

can be calculated to be $4/9$, independently of the marginal distributions. Consider the case when $X, Y \sim \mathcal{N}(0, 1)$. The boxplots of the ratio $\hat{\zeta}_1/\zeta_1$ are presented in Figure 5.6b. The control variates are built with $M = 5$ fully grown trees.

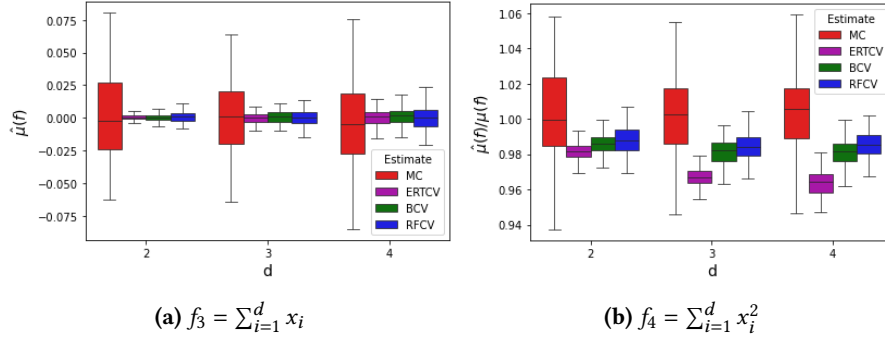


Figure 5.5: Boxplots of $\hat{\mu}(f_3)$ and $\hat{\mu}(f_4)/\mu(f_4)$ obtained over 100 replications for Gaussian mixture model

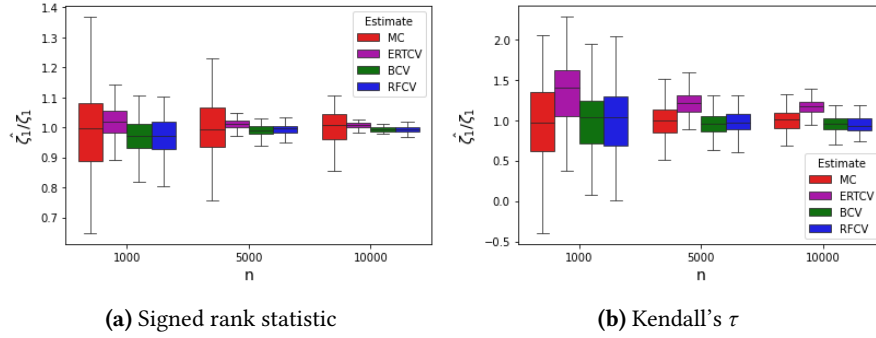


Figure 5.6: Boxplots of $\hat{\zeta}_1/\zeta_1$ obtained over 100 replications for U-statistics

For the Gaussian mixture model with $f_4 = \sum_{i=1}^d x_i^2$ (Figure 5.5b) and Kendall's τ (Figure 5.6b), we observe that the proposed estimates are biased. This phenomenon is discussed in Hastie et al. (2001) and Geurts et al. (2006). The bias of a random forest equals the bias of any of the individual sampled trees, which is typically greater than the bias of an unpruned tree due to randomization and reduced sample space. Therefore, the improvements in prediction obtained by Bagging, Random Forests, and Extremely Randomized Trees are primarily a result of variance reduction.

5.5 Appendix

In this section, we consider another type of random forest known as Mondrian forests and present additional numerical results for the control variate estimator based on this approach.

5.5.1 Mondrian forests

Mondrian forests are a class of efficient and accurate online random forests introduced by Lakshminarayanan et al. (2014). The algorithm for this class of forests is based on the Mondrian process (Roy and Teh, 2008), a natural probability distribution on the set of recursive partitions of a bounded domain. In comparison with classical random forests, the trees in the Mondrian forest algorithm differ in that the splits at internal nodes are sampled independently from the target variables, each node is associated with a specific time indicating when the split occurs, and the lifetime parameter $\lambda > 0$ controls how many splits are made, similarly to how the maximum depth works in standard regression trees.

The Mondrian forest algorithm (Mourtada et al., 2018)

Given a rectangular box $C = \prod_{j=1}^d [a_j, b_j] \subseteq \mathbb{R}^d$, where a_j and b_j represent the lower and upper bounds of the feature space in the j -th dimension, we let $|C| := \sum_{j=1}^d (b_j - a_j)$ denote its linear dimension. The Mondrian process $\text{MP}(C)$ is a distribution on (infinite) tree partitions of C . Mondrian partitions are built by iteratively splitting cells at some random time which depends on the linear dimension of the cell; the splitting probability on each side is proportional to the side length of the cell, and the position is drawn uniformly. The Mondrian process distribution $\text{MP}(\lambda, C)$ is a distribution on tree partitions of C , resulting from the pruning of partitions drawn from $\text{MP}(C)$. The pruning is done by removing all splits occurring after time $\lambda > 0$. Sampling from the distribution $\text{MP}(\lambda, C)$ can be done efficiently by applying the recursive procedure described in Algorithm 5.3.

Algorithm 5.3 $\text{SampleMondrian}(C, \tau, \lambda)$: sampling a Mondrian partition of C

Require: A cell $C = \prod_{j=1}^d [a_j, b_j]$, starting time τ and lifetime parameter λ

- 1: Sample a random variable $E_C \sim \text{Exp}(|C|)$
- 2: **if** $\tau + E_C \leq \lambda$ **then**
- 3: Sample a split dimension $J \in \{1, \dots, d\}$, with $P(J = j) = (b_j - a_j)/|C|$
- 4: Sample a split threshold S_J uniformly in $[a_J, b_J]$
- 5: Split C along the split (J, S_J) : let $C_0 = \{x \in C : x_J \leq S_J\}$ and $C_1 = C \setminus C_0$
- 6: **return** $\text{SampleMondrian}(C_0, \tau + E_C, \lambda) \cup \text{SampleMondrian}(C_1, \tau + E_C, \lambda)$
- 7: **else**
- 8: **return** C (i.e., do not split C)
- 9: **end if**

5.5.2 Additional numerical simulations

We perform additional simulation studies for the integrand $f_1(x_1, \dots, x_d) = \prod_{i=1}^d \log(2)2^{1-x_i} - 1$ with respect to the uniform distribution on $[0, 1]^d$, constructing control variates using Mondrian forests, which we call Mondrian Control Variates (MCV).

For building Mondrian trees, we use the implementation from the Python library *scikit-garden* (Kumar and contributors, 2017). The choice has fallen on this library because it is compatible with *scikit-learn* (Pedregosa et al., 2011), which we use to construct all other types of regression trees and to extract their boundaries when calculating the integrals of the control variates. The setup for the estimates is the same as in Section 5.4.

Maximum depth. Recall that we operate with only one tree. We fix the Monte Carlo sample size at $n = 10\,000$ and the dimension of the integration domain at $d = 2, 3, 4$. For Random Forests, we set $K = \lceil d/2 \rceil$. The RMSE plots for f_1 with different maximum depths are shown in Figure 5.7.

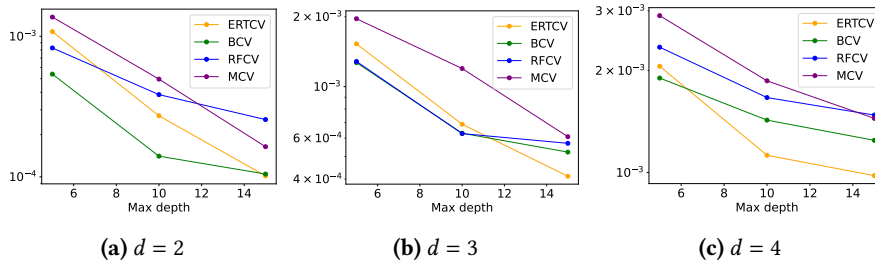


Figure 5.7: Plots of the root mean square errors for f_1 with different maximum depths obtained over 100 replications

As we can see, MCV does not outperform the other estimates. However, this is because maximum depth is not a tuning parameter for Mondrian trees. To obtain more reasonable results, we need to account for the lifetime parameter λ . However, the current implementation does not consider this. While the splits are performed using the Mondrian process, the stopping criteria remain the same as in classical regression trees, such as reaching a maximum tree depth or when the number of samples in a node falls below a minimum threshold.

Number of trees. Recall that we fix $d = 4$ and $n = 1\,000$. The maximum depth is taken as $\lceil \log(n) \rceil$. Figure 5.8 shows the results for f_1 with different numbers of trees.

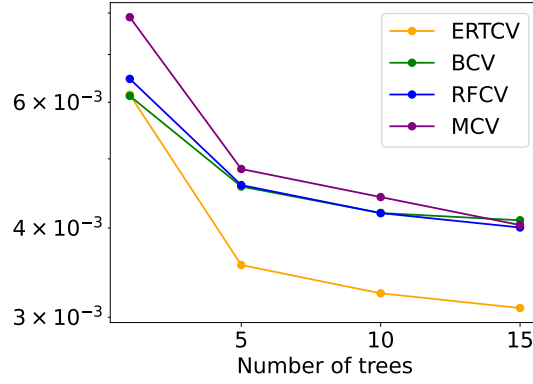


Figure 5.8: Plot of the root mean square errors for f_1 with different tree numbers obtained over 100 replications

We observe that the performance of MCV is nearly the same as that of BCV and RFCV. However, the theoretical investigations for a tuning parameter such as the number of trees for Mondrian forests are presented in Mourtada et al. (2018) and O'Reilly and Tran (2024). This parameter must be chosen in conjunction with the other tuning parameter, the lifetime parameter λ .

Optimal trees. Recall that we perform simulation studies for f_1 with $d = 2, 3, 4$ and $n = 100, 1000, 10000$. For the control variates, we build fully grown trees. The number of trees is chosen as $M = 5$. In Figure 5.9, the plots of the root mean square errors for f_1 are presented.

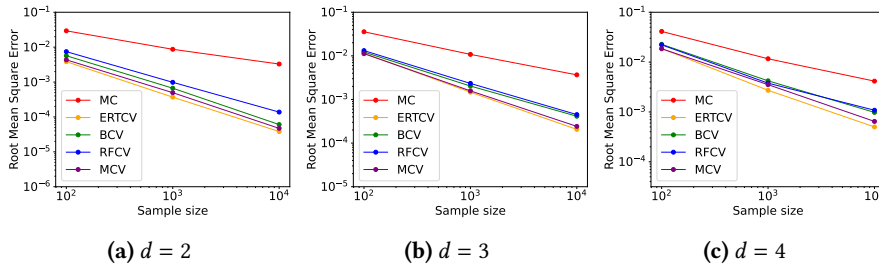


Figure 5.9: Plots of the root mean square errors for f_1 obtained over 100 replications with fully grown trees

We observe that MCV performs as well as the other estimates. However, as mentioned above, theoretical analyses of Mondrian forests (Mourtada et al., 2018; O'Reilly and Tran, 2024) suggest optimal values for both the number of trees and the lifetime parameter λ , which may improve both accuracy and computational efficiency.

Conclusion and discussion

6

The main objective of this thesis was to improve the accuracy of the standard Monte Carlo integration procedure using the variance reduction technique called the method of control variates. By definition, control variates are functions with known expectations. However, the main challenge was to construct suitable control variates for each specific problem to effectively reduce the variance of the integration procedure. During the research, we explored various techniques from mathematics and machine learning to construct such functions. Additionally, we investigated various applications of this method.

In Chapter 2, we explored another variance reduction technique called adaptive importance sampling and combined it with control variates. This approach presented both theoretical and practical challenges. Theoretical aspects included establishing a non-asymptotic error bound based on martingale theory due to the lack of independence between samples. Additionally, during the proof, we developed a novel bound on the smallest eigenvalue of specific random matrices. We also investigated the properties of this approach. On the practical side, we employed Stein control variates, which have the advantage that they do not require knowledge of the proportionality constant of the target density.

In Chapter 3, we examined a widely used machine learning technique, namely the nearest neighbor method, to construct control variates. This approach presented several challenges. We established the convergence rates of two proposed estimates on a bounded metric space. Additionally, we considered closed Riemannian manifolds as a particular case, since later in the numerical part, we performed integration on such manifolds as the unit sphere and the orthogonal group. However, the non-asymptotic bounds were established only for a compact subset of Euclidean space. Besides, a bound on the L_p approximation of the nearest neighbor function estimate was derived. A practical issue was calculating the expectation of the nearest neighbor estimate, which required computation of the Voronoi volumes. This task is difficult to accomplish because Voronoi cells rapidly increase in complexity as the dimension grows. We addressed this issue by approximating the required expectation without extra evaluations of the integrand. Regarding possible extensions, the method can be practically applied to other sampling designs,

such as MCMC and adaptive importance sampling, even though the theory for this method is restricted to independent random variables.

In Chapter 4, we focused on incorporating control variates for a specific problem originating from optimal transport and widely used in machine learning, namely the calculation of the Sliced-Wasserstein distance. Since the integration domain is the unit sphere, we proposed employing spherical harmonics as control variates. The theoretical analysis was complicated by the limited knowledge of the smoothness properties of the integrand appearing in the definition of the Sliced-Wasserstein distance; currently, its Lipschitz continuity is the only established property. We derived the asymptotic integration error bound of the proposed estimator. However, the concentration inequality remained an open question. From a practical perspective, we implemented our estimate in the Kernel SVM for image classification. A potential practical issue can be that the number of control variates increases rapidly with the dimension. However, in such cases, one could rely on the lasso procedure for control variate selection, as discussed in Leluc et al. (2021).

In Chapter 5, we explored another machine learning technique, specifically random forests, for constructing control variates. Among the variety of different random forests, we focused mostly on three types: Bagging, Random Forests, and Extremely Randomized Trees. We performed numerical analysis to investigate how different tuning parameters for constructing trees influenced the performance of the proposed estimates. However, the theoretical analysis of the proposed estimates remained an open problem due to insufficient theoretical investigations on these random forest methods. From a practical perspective, we applied our method to calculate the variance of the asymptotic distribution of U-statistics. Concerning the drawbacks, the method is limited to cases of product probability measures, and it introduces bias in some situations.

For all proposed estimates, the question of inference was not discussed. However, the asymptotic bound for the SHCV estimator in Chapter 4 was derived based on Theorem 1 from Portier and Segers (2019). This paper establishes also the central limit theorem for the estimator (see their Theorem 2). This justifies the use of the usual asymptotic confidence intervals with nominal coverage $1 - \alpha$ given by $\hat{\mu}_{n,ols} \pm z_{1-\alpha/2} \hat{\sigma}_{n,ols}$, where z_p is the p -th quantile of the standard normal distribution, $\hat{\mu}_{n,ols}$ corresponds to the SHCV estimate, and $\hat{\sigma}_{n,ols}^2$ is the usual estimator of the residual variance in the multiple linear regression problem that underlies the OLSMC estimator.

Concerning the AISCV estimator in Chapter 2, it may be possible to establish its asymptotic normality, similar to the normalized AIS estimator in Portier and Delyon (2018). First, in their Theorem 1, the authors establish a central limit theorem for the AIS estimator using the martingale CLT (Hall

and Heyde, 1980, Corollary 3.1). Then, in their Corollary 1, they apply the delta method to extend the result to the normalized version of the estimator.

For the control variate estimators based on nearest neighbor (Chapter 3) or random forest (Chapter 5) methods, we can construct confidence intervals by means of Monte Carlo simulations. Recall that we construct the control variate estimate in the following form:

$$I_n^{(cv)}(f) = \frac{1}{n} \sum_{i=1}^n \left[f(X_i) - \left\{ \hat{f}_n(X_i) - I(\hat{f}_n) \right\} \right].$$

To avoid additional evaluations of the integrand, we propose drawing new N samples only for the control variate part. After generating multiple sets of control variate estimates via Monte Carlo simulations, we can construct confidence intervals based on the empirical distribution of these estimates. Specifically, we calculate the quantiles of the distribution of $I_n^{(cv)}$, which provides a range of values within which the true integral lies with a given confidence level.

Among the possible lines of research could be the following:

- In South et al. (2023), the authors present empirical evidence that the use of penalized regression techniques such as lasso and ridge regression in the selection of high-dimensional control variates offers performance gains over the classical least squares method. Their findings indicate that lasso generally outperforms ridge regression in the considered examples. While Leluc et al. (2021) provide theoretical support for lasso-based control variate selection, research on ridge regression in this context is lacking. Additionally, it is viable to explore the combination of these techniques, known as the elastic net (Zou and Hastie, 2005). Moreover, more advanced methods like the adaptive lasso (Zou, 2006) and the adaptive elastic net (Zou and Zhang, 2009) could also be considered.
- In Chapter 5, we considered classical ensembles of regression trees and, in its appendix, we slightly explored more advanced random forests, namely Mondrian forests (Lakshminarayanan et al., 2014), for constructing control variates. A deeper investigation into the latter could be promising, as some theoretical studies exist. For instance, Mourtada et al. (2018) and O'Reilly and Tran (2024) provide minimax optimal rates for this class of forests in arbitrary dimensions.

Bibliography

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). TensorFlow: A system for Large-Scale machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pages 265–283.
- Achddou, J., Lam-Weil, J., Carpentier, A., and Blanchard, G. (2019). A minimax near-optimal algorithm for adaptive rejection sampling. In *Algorithmic Learning Theory*, pages 94–126. PMLR.
- Alpaydin, E. and Kaynak, C. (1998). Optical Recognition of Handwritten Digits. UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C50P49>.
- Atkinson, K. and Han, W. (2012). *Spherical Harmonics and Approximations on the Unit Sphere: An Introduction*. Lecture Notes in Mathematics. Springer.
- Audibert, J.-Y. and Tsybakov, A. B. (2007). Fast learning rates for plug-in classifiers. *Ann. Statist.*, 35(2):608–633.
- Bakhvalov, N. S. (2015). On the approximate calculation of multiple integrals. *J. Complex.*, 31(4):502–516.
- Bardenet, R. and Hardy, A. (2020). Monte carlo with determinantal point processes. *Ann. Appl. Probab.*, 30(1):368–417.
- Basri, R. and Jacobs, D. W. (2003). Lambertian reflectance and linear subspaces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(2):218–233.
- Bayraktar, E. and Guo, G. (2021). Strong equivalence between metrics of Wasserstein type. *Electron. Commun. Probab.*, 26(none):1 – 13.
- Belomestny, D., Iosipoi, L., Paris, Q., and Zhivotovskiy, N. (2022). Empirical variance minimization with applications in variance reduction and optimal control. *Bernoulli*, 28(2):1382–1407.
- Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *CACM*, 18(9):509–517.

- Bhatia, R., Jain, T., and Lim, Y. (2019). On the Bures–Wasserstein distance between positive definite matrices. *Expo. Math.*, 37(2):165–191.
- Biau, G. (2012). Analysis of a random forests model. *J. Mach. Learn. Res.*, 13:1063–1095.
- Biau, G. and Devroye, L. (2015). *Lectures on the Nearest Neighbor Method*. Springer International Publishing.
- Black, F. and Scholes, M. (1973). The pricing of options and corporate liabilities. *J. Political Econ.*, 81(3):637–654.
- Blanchet, J., Chen, H., Lu, Y., and Ying, L. (2023). When can regression-adjusted control variates help? In *Adv. Neural Inf. Process. Syst.*, volume 36, pages 36566–36578.
- Bonneel, N., Rabin, J., Peyré, G., and Pfister, H. (2015). Sliced and Radon Wasserstein barycenters of measures. *J. Math. Imaging Vision*, 51:22–45.
- Bonnotte, N. (2013). *Unidimensional and evolution methods for optimal transportation*. PhD thesis, Université Paris Sud-Paris XI; Scuola normale superiore (Pise, Italie).
- Boucheron, S., Lugosi, G., and Massart, P. (2013). *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press.
- Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24:123–140.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45:5–32.
- Breiman, L. (2004). Consistency for a simple model of random forests. Technical report, University of California, Berkeley.
- Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). *Classification and Regression Trees*. Chapman and Hall/CRC.
- Caflich, R. E. (1998). Monte Carlo and quasi-Monte Carlo methods. *Acta Numer.*, 7:1–49.
- Cappé, O., Douc, R., Guillin, A., Marin, J.-M., and Robert, C. P. (2008). Adaptive importance sampling in general mixture classes. *Stat. Comput.*, 18(4):447–459.
- Cappé, O., Guillin, A., Marin, J.-M., and Robert, C. P. (2004). Population Monte Carlo. *J. Comput. Graph. Stat.*, 13(4):907–929.

- Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al. (2015). Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*.
- Chopin, N. (2004). Central limit theorem for sequential Monte Carlo methods and its application to bayesian inference. *Ann. Statist.*, 32(6):2385–2411.
- Chopin, N. and Gerber, M. (2022). Higher-order stochastic integration through cubic stratification. *arXiv preprint arXiv:2210.01554*.
- Coeurjolly, J.-F., Mazoyer, A., and Amblard, P.-O. (2021). Monte Carlo integration of non-differentiable functions on $[0, 1]^d$, $d = 1, \dots, d$, using a single determinantal point pattern defined on $[0, 1]^d$. *Electron. J. Statist.*, 15(2):6228–6280.
- Cohen, T. S., Geiger, M., Köhler, J., and Welling, M. (2018). Spherical CNNs. In *International Conference on Learning Representations*.
- Combes, R. (2015). An extension of McDiarmid’s inequality. *arXiv preprint arXiv:1511.05240*.
- Courty, N., Flamary, R., Habrard, A., and Rakotomamonjy, A. (2017). Joint distribution optimal transportation for domain adaptation. *Adv. Neural Inf. Process. Syst.*, 30:3733–3742.
- Craven, P. and Wahba, G. (1978). Smoothing noisy data with spline functions. *Numer. Math.*, 31(4):377–403.
- Cuturi, M. (2013). Sinkhorn distances: Lightspeed computation of optimal transport. In *Adv. Neural Inf. Process. Syst.*, volume 26, pages 2292–2300.
- Dai, B., He, N., Dai, H., and Song, L. (2016). Provable Bayesian inference via particle mirror descent. In *Artificial Intelligence and Statistics*, pages 985–994. PMLR.
- Dai, F. (2013). *Approximation theory and harmonic analysis on spheres and balls*. Springer.
- Del Moral, P., Doucet, A., and Jasra, A. (2006). Sequential Monte Carlo samplers. *J. R. Stat. Soc. Ser. B Stat. Method.*, 68(3):411–436.
- Delyon, B. and Portier, F. (2021). Safe adaptive importance sampling: A mixture approach. *Ann. Statist.*, 49(2):885–917.
- Deshpande, I., Hu, Y.-T., Sun, R., Pyrros, A., Siddiqui, N., Koyejo, S., Zhao, Z., Forsyth, D., and Schwing, A. G. (2019). Max-sliced Wasserstein distance and its use for gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10648–10656.

- Deshpande, I., Zhang, Z., and Schwing, A. G. (2018). Generative modeling using the sliced Wasserstein distance. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3483–3491.
- Diaconis, P. and Evans, S. N. (2001). Linear functionals of eigenvalues of random matrices. *Trans. Amer. Math. Soc.*, 353(7):2615–2633.
- Douc, R. and Moulines, E. (2008). Limit theorems for weighted samples with applications to sequential Monte Carlo methods. *Ann. Statist.*, pages 2344–2376.
- Doucet, A., De Freitas, N., Gordon, N. J., et al. (2001). *Sequential Monte Carlo Methods in Practice*. Number 2. Springer, New York.
- Dowson, D. and Landau, B. (1982). The Fréchet distance between multivariate normal distributions. *J. Multivar. Anal.*, 12(3):450–455.
- Dua, D. and Graff, C. (2019). Uci Machine Learning Repository [<http://archive.ics.uci.edu/ml>]. irvine, ca: University of california. *School of Information and Computer Science*, 25:27.
- Dutordoir, V., Durrande, N., and Hensman, J. (2020). Sparse Gaussian processes with spherical harmonic features. In *International Conference on Machine Learning*, pages 2793–2802. PMLR.
- Eckhardt, R. (1987). Stan Ulam, John von Neumann, and the Monte Carlo method. *Los Alamos Science*, 15.
- Evans, M. and Swartz, T. (2000). *Approximating Integrals via Monte Carlo and Deterministic Methods*. Oxford Statistical Science Series. Oxford University Press, Oxford.
- Feragen, A., Lauze, F., and Hauberg, S. (2015). Geodesic exponential kernels: When curvature and linearity conflict. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3032–3042.
- Friedman, J. H., Bentley, J. L., and Finkel, R. A. (1977). An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Softw.*, 3(3):209–226.
- Gadat, S., Klein, T., and Marteau, C. (2016). Classification in general finite dimensional spaces with the k -nearest neighbor rule. *Ann. Statist.*, 44(3):982–1009.
- Gautschi, W. (2004). *Orthogonal polynomials: computation and approximation*. OUP Oxford.

- Genevay, A., Peyré, G., and Cuturi, M. (2018). Learning generative models with Sinkhorn divergences. In *Int. Conf. Artif. Intell. Stat.*, pages 1608–1617. PMLR.
- Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, 63:3–42.
- Geweke, J. (1989). Bayesian inference in econometric models using Monte Carlo integration. *Econometrica: Journal of the Econometric Society*, pages 1317–1339.
- Glasserman, P. (2004). *Monte Carlo Methods in Financial Engineering*. Springer, New York, NY, USA.
- Glynn, P. W. and Szechtman, R. (2002). Some new perspectives on the method of control variates. In *Monte Carlo and Quasi-Monte Carlo Methods 2000*, pages 27–49. Springer.
- Gorham, J. and Mackey, L. (2015). Measuring sample quality with Stein’s method. *Adv. Neural Inf. Process. Syst.*, 28.
- Green, R. (2003). Spherical harmonic lighting: The gritty details. In *Archives of the game developers conference*, volume 56, page 4.
- Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. (2017). Improved training of Wasserstein GANs. In *Adv. Neural Inf. Process. Syst.*, volume 30, page 5769–5779.
- Haber, S. (1966). A modified Monte-Carlo quadrature. *Math. Comp.*, 20(95):361–368.
- Haber, S. (1967). A modified Monte-Carlo quadrature. II. *Math. Comp.*, 21(99):388–397.
- Haber, S. (1969). Stochastic quadrature formulas. *Math. Comp.*, 23(108):751–764.
- Hall, P. and Heyde, C. (1980). *Martingale Limit Theory and Its Application*. Communication and Behavior. Academic Press.
- Han, R. (2023). Sliced Wasserstein distance between probability measures on Hilbert spaces. arXiv preprint arXiv:2307.05802.
- Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA.

- Hesterberg, T. (1995). Weighted average importance sampling and defensive mixture distributions. *Technometrics*, 37(2):185–194.
- Heston, S. L. (1993). A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The review of financial studies*, 6(2):327–343.
- Hickernell, F. J., Lemieux, C., and Owen, A. B. (2005). Control variates for Quasi-Monte Carlo. *Stat. Sci.*, 20(1):1–31.
- Higdon, D., McDonnell, J. D., Schunck, N., Sarich, J., and Wild, S. M. (2015). A Bayesian approach for parameter estimation and prediction using a computationally intensive model. *J. Phys. G Nucl. Part. Phys.*, 42(3):034009.
- Hinrichs, A., Krieg, D., Novak, E., Prochno, J., and Ullrich, M. (2020). On the power of random information. *Multivariate Algorithms and information-based complexity*, 27:43–64.
- Hoff III, K. E., Keyser, J., Lin, M., Manocha, D., and Culver, T. (1999). Fast computation of generalized Voronoi diagrams using graphics hardware. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 277–286.
- Hoffman, M. D., Gelman, A., et al. (2014). The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623.
- Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Comput. Sci. Eng.*, 9(03):90–95.
- Ichida, R. (2000). The injectivity radius and the fundamental group of compact homogeneous riemannian manifolds of positive curvature. *Tsukuba J. Math.*, 24(1):139–156.
- Jie, T. and Abbeel, P. (2010). On a connection between importance sampling and the likelihood ratio policy gradient. In *Adv. Neural Inf. Process. Syst.*, volume 23. Curran Associates, Inc.
- Jin, C., Netrapalli, P., Ge, R., Kakade, S. M., and Jordan, M. I. (2019). A short note on concentration inequalities for random vectors with subgaussian norm. *arXiv preprint arXiv:1902.03736*.
- Johnson, J., Douze, M., and Jégou, H. (2019). Billion-scale similarity search with GPUs. *IEEE Trans. Big Data*, 7(3):535–547.
- Jourdain, B. (2009). Adaptive variance reduction techniques in finance. *Radon Series Comp. Appl. Math.*, 8:1–18.

- Kabatyanskii, G. and Levenshtein, V. (1974). Bounds for packings on a sphere and in space. *Problems of Information Transmission*, 95:148–158.
- Kawai, R. (2020). Adaptive importance sampling and control variates. *J. Math. Anal. Appl.*, 483(1):123608.
- Kloek, T. and Van Dijk, H. K. (1978). Bayesian estimates of equation system parameters: an application of integration by Monte Carlo. *Econometrica: Journal of the Econometric Society*, pages 1–19.
- Klusowski, J. (2021). Sharp analysis of a simple model for random forests. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, Proceedings of Machine Learning Research, pages 757–765.
- Kohler, M. and Krzyżak, A. (2013). Optimal global rates of convergence for interpolation problems with random design. *Statist. & Probabil. Lett.*, 83(8):1871–1879.
- Kohler, M., Krzyżak, A., and Walk, H. (2006). Rates of convergence for partitioning and nearest neighbor regression estimates with unbounded data. *J. Multivariate Anal.*, 97(2):311–323.
- Kokarev, G. (2021). Berger’s inequality in the presence of upper sectional curvature bound. *Int. Math. Res. Not.*, 2022(15):11262–11303.
- Kolouri, S., Nadjahi, K., Simsekli, U., Badeau, R., and Rohde, G. (2019). Generalized sliced Wasserstein distances. *Adv. Neural Inf. Process. Syst.*, 32.
- Kolouri, S., Pope, P. E., Martin, C. E., and Rohde, G. K. (2018). Sliced Wasserstein auto-encoders. In *International Conference on Learning Representations*.
- Kolouri, S., Zou, Y., and Rohde, G. K. (2016). Sliced Wasserstein kernels for probability distributions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5258–5267.
- Korba, A. and Portier, F. (2022). Adaptive importance sampling meets mirror descent: a bias-variance tradeoff. In *International Conference on Artificial Intelligence and Statistics*, pages 11503–11527. PMLR.
- Krieg, D. and Novak, E. (2017). A universal algorithm for multivariate integration. *Found. Comput. Math.*, 17(4):895–916.
- Krieg, D., Novak, E., and Sonnleitner, M. (2022). Recovery of Sobolev functions restricted to iid sampling. *Math. Comp.*, 91(338):2715–2738.

- Kumar, M. and contributors (2017). Scikit-garden: A garden for scikit-learn compatible trees. <https://github.com/scikit-garden/scikit-garden>.
- Lakshminarayanan, B., Roy, D., and Teh, Y. (2014). Mondrian forests: Efficient online random forests. *Adv. Neural Inf. Process. Syst.*, 4.
- Lee, J. (2019). *Introduction to Riemannian Manifolds*. Graduate Texts in Mathematics. Springer International Publishing.
- Leluc, R., Dieuleveut, A., Portier, F., Segers, J., and Zhuman, A. (2024a). Sliced-Wasserstein estimation with spherical harmonics as control variates. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 27191–27214. PMLR.
- Leluc, R., Portier, F., and Segers, J. (2021). Control variate selection for Monte Carlo integration. *Stat. Comput.*, 31.
- Leluc, R., Portier, F., Segers, J., and Zhuman, A. (2022). A Quadrature Rule combining Control Variates and Adaptive Importance Sampling. In *Adv. Neural Inf. Process. Syst.*, volume 35, pages 11842–11853. Curran Associates, Inc.
- Leluc, R., Portier, F., Segers, J., and Zhuman, A. (2024b). Speeding up Monte Carlo integration: Control neighbors for optimal convergence. *Bernoulli*, in press.
- Liu, H., Feng, Y., Mao, Y., Zhou, D., Peng, J., and Liu, Q. (2017). Action-dependent control variates for policy optimization via Stein’s identity. *arXiv preprint arXiv:1710.11198*.
- Liutkus, A., Simsekli, U., Majewski, S., Durmus, A., and Stöter, F.-R. (2019). Sliced-Wasserstein flows: Nonparametric generative modeling via optimal transport and diffusions. In *International Conference on Machine Learning*, pages 4104–4113. PMLR.
- Louppe, G. (2014). *Understanding Random Forests: From Theory to Practice*. PhD dissertation, University of Liège.
- Martino, L., Elvira, V., Luengo, D., and Corander, J. (2017). Layered adaptive importance sampling. *Stat. Comput.*, 27(3):599–623.
- Merton, R. C. (1973). Theory of rational option pricing. *Bell J. Econ. Manage. Sci.*, pages 141–183.

- Metropolis, N. (1987). The Beginnings of the Monte Carlo Method. *Los Alamos Science*, 15.
- Meunier, D., Pontil, M., and Ciliberto, C. (2022). Distribution regression with sliced Wasserstein kernels. In *International Conference on Machine Learning*, pages 15501–15523. PMLR.
- Mezzadri, F. (2007). How to generate random matrices from the classical compact groups. *Not. Am. Math. Soc.*, 54(5):592–604.
- Mira, A., Solgi, R., and Imparato, D. (2013). Zero variance Markov Chain Monte Carlo for Bayesian estimators. *Stat. Comput.*, 23(5):653–662.
- Molenberghs, G. and Verbeke, G. (2000). *Linear Mixed Models for Longitudinal Data*. Springer New York, New York, NY.
- Mourtada, J., Gaïffas, S., and Scornet, E. (2018). Minimax optimal rates for mondrian trees and forests. *Ann. Statist.*, 48.
- Nadjahi, K., De Bortoli, V., Durmus, A., Badeau, R., and Şimşekli, U. (2020a). Approximate Bayesian computation with the sliced-wasserstein distance. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5470–5474. IEEE.
- Nadjahi, K., Durmus, A., Chizat, L., Kolouri, S., Shahrampour, S., and Simsekli, U. (2020b). Statistical and topological properties of sliced probability divergences. *Adv. Neural Inf. Process. Syst.*, 33:20802–20812.
- Nadjahi, K., Durmus, A., Jacob, P. E., Badeau, R., and Simsekli, U. (2021). Fast approximation of the sliced-wasserstein distance using concentration of random projections. *Adv. Neural Inf. Process. Syst.*, 34:12411–12424.
- Newman, D. J. and Shapiro, H. S. (1964). Jackson’s Theorem in Higher Dimensions. In *Conference on Approximation Theory (Oberwolfach, 1963)*, pages 208–219.
- Newton, N. J. (1994). Variance reduction for simulated diffusions. *SIAM J. Appl. Math.*, 54(6):1780–1805.
- Nguyen, K., Bariletto, N., and Ho, N. (2024). Quasi-Monte Carlo for 3D Sliced Wasserstein. In *The Twelfth International Conference on Learning Representations*.
- Nguyen, K. and Ho, N. (2024). Sliced Wasserstein Estimation with Control Variates. In *The Twelfth International Conference on Learning Representations*.

- Nietert, S., Goldfeld, Z., Sadhu, R., and Kato, K. (2022). Statistical, robustness, and computational guarantees for sliced Wasserstein distances. *Adv. Neural Inf. Process. Syst.*, 35:28179–28193.
- Novak, E. (1988). *Deterministic and Stochastic Error Bounds in Numerical Analysis*. Springer, Berlin, Heidelberg.
- Novak, E. (2016). Some results on the complexity of numerical integration. In *Monte Carlo and Quasi-Monte Carlo Methods*, pages 161–183. Springer.
- Oates, C. J., Cockayne, J., Briol, F.-X., and Girolami, M. (2019). Convergence rates for a class of estimators based on Stein’s method. *Bernoulli*, 25(2):1141–1159.
- Oates, C. J., Girolami, M., and Chopin, N. (2017). Control functionals for Monte Carlo integration. *J. R. Stat. Soc. Ser. B Stat. Method.*, 79(3):695–718.
- Oh, M.-S. and Berger, J. O. (1992). Adaptive importance sampling in monte carlo integration. *J. Stat. Comput. Simul.*, 41(3-4):143–168.
- Omohundro, S. M. (1989). *Five balltree construction algorithms*. International Computer Science Institute Berkeley.
- O’Reilly, E. and Tran, N. M. (2024). Minimax rates for high-dimensional random tessellation forests. *J. Mach. Learn. Res.*, 25(89):1–32.
- Owen, A. and Zhou, Y. (2000). Safe and effective importance sampling. *J. Amer. Statist. Assoc.*, 95(449):135–143.
- Owen, A. B. (2013). *Monte Carlo theory, methods and examples*. <https://artowen.su.domains/mc/>.
- Pastur, L. and Vasilchuk, V. (2004). On the moments of traces of matrices of classical groups. *Comm. Math. Phys.*, 252(1-3):149–166.
- Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in pytorch.
- Patil, A., Huard, D., and Fonnesbeck, C. J. (2010). Pymc: Bayesian stochastic modelling in python. *J. Stat. Softw.*, 35(4):1.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.*, 12:2825–2830.

- Peyré, G., Cuturi, M., et al. (2019). Computational optimal transport: With applications to data science. *Found. Trends Mach. Learn.*, 11(5-6):355–607.
- Plassier, V., Portier, F., and Segers, J. (2020). Risk bounds when learning infinitely many response functions by ordinary linear regression. *arXiv preprint arXiv:2006.09223*. To appear in *Annales de l'Institut Henri Poincaré (B) Probabilités et Statistiques*.
- Portier, F. (2023). Nearest neighbor process: weak convergence and non-asymptotic bound. *arXiv preprint arXiv:2110.15083*.
- Portier, F. and Delyon, B. (2018). Asymptotic optimality of adaptive importance sampling. *Adv. Neural Inf. Process. Syst.*, 31.
- Portier, F. and Segers, J. (2018). On the weak convergence of the empirical conditional copula under a simplifying assumption. *J. Multivariate Anal.*, 166:160–181.
- Portier, F. and Segers, J. (2019). Monte Carlo integration with a growing number of control variates. *J. Appl. Probab.*, 56(4):1168–1186.
- Rabin, J., Peyré, G., Delon, J., and Bernot, M. (2012). Wasserstein barycenter and its application to texture mixing. In *Scale Space and Variational Methods in Computer Vision: Third International Conference, SSVM 2011, Ein-Gedi, Israel, May 29–June 2, 2011, Revised Selected Papers 3*, pages 435–446. Springer.
- Ragozin, D. L. (1971). Constructive Polynomial Approximation on Spheres and Projective Spaces. *Trans. Amer. Math. Soc.*, 162:157–170.
- Ramamoorthi, R. and Hanrahan, P. (2001). An efficient representation for irradiance environment maps. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 497–500.
- Richards, F. M. (1974). The interpretation of protein structures: total volume, group volume distributions and packing density. *J. Mol. Biol.*, 82(1):1–14.
- Roy, D. and Teh, Y. (2008). The mondrian process. *Adv. Neural Inf. Process. Syst.*, 21:1377–1384.
- Rubinstein, R. Y. (1981). *Simulation and the Monte Carlo method*. John Wiley & Sons, Inc., New York.
- Rycroft, C. (2009). Voro++: A three-dimensional Voronoi cell library in C++. Technical report, Lawrence Berkeley National Lab.(LBNL), Berkeley, CA (United States).

- Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989). Design and analysis of computer experiments. *Statist. Sci.*, 4(4):409–423.
- South, L. F., Oates, C. J., Mira, A., and Drovandi, C. (2023). Regularized zero-variance control variates. *Bayesian Anal.*, 18(3):865–888.
- Stein, C. (1972). A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. In *Proceedings of the sixth Berkeley symposium on mathematical statistics and probability, volume 2: Probability theory*, volume 6, pages 583–603. University of California Press.
- Stein, E. M. and Shakarchi, R. (2005). *Real Analysis: Measure Theory, Integration, and Hilbert Spaces*. Princeton University Press, Princeton.
- Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 36(2):111–133.
- Szabó, Z., Sriperumbudur, B. K., Póczos, B., and Gretton, A. (2016). Learning theory for distribution regression. *J. Mach. Learn. Res.*, 17(1):5272–5311.
- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 58(1):267–288.
- Toscano-Palmerin, S. and Frazier, P. I. (2022). Bayesian optimization with expensive integrands. *SIAM J. Optim.*, 32(2):417–444.
- Tropp, J. (2015). An introduction to matrix concentration inequalities. arXiv:1501.01571.
- van der Vaart, A. W. (1998). *Asymptotic Statistics*. Cambridge University Press, Cambridge.
- Villani, C. (2009). *Optimal Transport: Old and New*. Springer, Berlin, Heidelberg.
- Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods*, 17(3):261–272.
- Wan, R., Zhong, M., Xiong, H., and Zhu, Z. (2020). Neural control variates for monte carlo variance reduction. In *Machine Learning and Knowledge Discovery in Databases*, pages 533–547. Springer International Publishing.

- Wang, C., Chen, X., Smola, A., and Xing, E. (2013). Variance reduction for stochastic gradient optimization. In *Adv. Neural Inf. Process. Syst.*, volume 26. Curran Associates, Inc.
- Zeger, K. and Gersho, A. (1994). Number of nearest neighbors in a Euclidean code. *IEEE Trans. Inf. Theory*, 40(5):1647–1649.
- Zhang, P. (1996). Nonparametric importance sampling. *J. Amer. Statist. Assoc.*, 91(435):1245–1253.
- Zhang, T. (2004). Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the twenty-first international conference on Machine learning*, page 116.
- Zimmermann, R. and Stoye, J. (2024). High curvature means low-rank: On the sectional curvature of grassmann and stiefel manifolds and the underlying matrix trace inequalities. *arXiv preprint arXiv:2403.01879*.
- Zou, H. (2006). The adaptive lasso and its oracle properties. *J. Amer. Statist. Assoc.*, 101:1418–1429.
- Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Ser. B Stat. Method.*, 67(2):301–320.
- Zou, H. and Zhang, H. (2009). On the adaptive elastic-net with a diverging number of parameters. *Ann. Statist.*, 37:1733–1751.