where

$$\hat{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{\Sigma}^{-1/2}(\boldsymbol{\theta})\boldsymbol{A} \\ \boldsymbol{Q}^{1/2}(\boldsymbol{\theta}) \end{bmatrix}, \quad \hat{\boldsymbol{y}} = \begin{bmatrix} \boldsymbol{\Sigma}^{-1/2}(\boldsymbol{\theta})\boldsymbol{y} \\ \boldsymbol{Q}^{1/2}(\boldsymbol{\theta})\boldsymbol{\mu} \end{bmatrix} \quad [17]. \tag{2.27}$$

A sample $\boldsymbol{x}_i$ can be computed by minimising the following equation with respect to $\hat{\boldsymbol{x}}$:

$$\boldsymbol{x}_i = \arg\min_{\hat{\boldsymbol{x}}} \|\hat{\boldsymbol{A}}\hat{\boldsymbol{x}} - (\hat{\boldsymbol{y}} + \boldsymbol{b})\|^2, \quad \boldsymbol{b} \sim \mathcal{N}(\boldsymbol{0}, \mathbf{I}), \tag{2.28}$$

where we add a randomised perturbation $\boldsymbol{b}$. Similar to Section 2.4, this expression can be rewritten as

$$\left(\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta})\boldsymbol{A} + \boldsymbol{Q}(\boldsymbol{\theta})\right)\boldsymbol{x}_i = \boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta})\boldsymbol{y} + \boldsymbol{Q}(\boldsymbol{\theta})\boldsymbol{\mu} + \boldsymbol{v}_1 + \boldsymbol{v}_2, \tag{2.29}$$

where the term $-\hat{\boldsymbol{A}}^T\boldsymbol{b}$ is decomposed as $\boldsymbol{v}_1 + \boldsymbol{v}_2$, with $\boldsymbol{v}_1 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta})\boldsymbol{A})$ and $\boldsymbol{v}_2 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}(\boldsymbol{\theta}))$, representing independent Gaussian random variables [6, 16].

If the Markov chain over the marginal posterior $\pi(\boldsymbol{\theta}|\boldsymbol{y})$ is ergodic, and the conditional samples $\boldsymbol{x}^{(k)} \sim \pi(\boldsymbol{x}|\boldsymbol{\theta}^{(k)}, \boldsymbol{y})$ are drawn independently, then the resulting joint chain $\{(\boldsymbol{x}, \boldsymbol{\theta})^{(1)}, \dots, (\boldsymbol{x}, \boldsymbol{\theta})^{(N)}\} \sim \pi(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{y})$ is also ergodic [18].

2.5.3 t-walk sampler as black box

If the parameters $\boldsymbol{x}$ are functionally dependent on the hyper-parameters $\boldsymbol{\theta}$, i.e., $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\theta})$, we can sample directly from the marginal posterior $\pi(\boldsymbol{\theta}|\boldsymbol{y})$ using the t-walk algorithm by Christen and Fox [19]. The t-walk is employed as a black-box sampler, requiring only the specification of the number of samples, burn-in period, support region, and the sampling distribution. Convergence to the target distribution is guaranteed by construction of the algorithm.

2.6 Numerical Approxiamtion Methods - Tensor Train

First, we provide a short overview of probability spaces and their associated measures, as a foundation for deriving marginal probability distribution, and then we give a brief introduction to the tensor train format. The motivation to use the tensor train format is that we can approximate a d-dimensional grid with far fewer data points compared to the total number of grid points.

Assume that the triple $(\Omega, \mathcal{F}, \mathbb{P})$ defines a probability space, where Ω denotes the complete sample space, $\mathcal{F}$ is a σ-algebra consisting of a collection of countable subsets $\{A_n\}_{n\in\mathbb{N}}$ with $A_n \subseteq \Omega$, and $\mathbb{P}$ is a probability measure defined on $\mathcal{F}$. The formal conditions for $\mathbb{P}$ to be a probability measure, and for $\mathcal{F}$ to be a σ-algebra over Ω, are given in Appendix C. We denote

$$\mathbb{P}(A) = \int_A \mathrm{d}\mathbb{P} \tag{2.30}$$

as the probability of an event $A \in \mathcal{F}$. By applying the Radon-Nikodym theorem [20], we can change variables

$$\mathbb{P}(A) = \int_A \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}x}\,\mathrm{d}x = \int_A \pi(x)\,\mathrm{d}x, \tag{2.31}$$

where $\mathrm{d}x$ is a reference measure on the same probability space, commonly referred to as the Lebesgue measure. The Radon-Nikodym derivative $\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}x}$ of $\mathbb{P}$ with respect to x, and is often interpreted as the probability density function (PDF) $\pi(x)$. Thus, we say that $\mathbb{P}$ has a density $\pi(x)$ with respect to x [21, Chapter 10].

Now, let $X : \Omega \longrightarrow \mathbb{R}^d$ be a d-dimensional random variable mapping from the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ to the measurable space $(\mathbb{R}^d, \mathcal{X})$, where $\mathcal{X}$ is a collection of subsets in $\mathbb{R}^d$. Then the associated PDF $\pi(x)$, is a joint density of X, induced by the probability measure on Ω [20, 22]. As by Cui et al. [23], we can define the parameter space as the Cartesian product $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_d$ with $x_k \in \mathcal{X}_k \subseteq \mathbb{R}$ and $x = (x_1, \ldots, x_k, \ldots, x_d)$. The marginal density function for the k-th component is then given by

$$f_{X_k}(x_k) = \int_{\mathcal{X}_1} \cdots \int_{\mathcal{X}_d} \lambda(x)\,\pi(x)\,\mathrm{d}x_1 \cdots \mathrm{d}x_{k-1}\,\mathrm{d}x_{k+1} \cdots \mathrm{d}x_d, \tag{2.32}$$

where we integrate over all dimensions except the k-th. Here, we introduce a weight function $\lambda(x)$ [24], which can be useful for quadrature rules??. Cui et al. [23] refer to $\lambda(x)$ as a "product-form Lebesgue-measurable weighting function" and define it as

$$\lambda(\mathcal{X}) = \prod_{i=1}^{d} \lambda_i(\mathcal{X}_i), \quad \text{where} \quad \lambda_i(\mathcal{X}_i) = \int_{\mathcal{X}_i} \lambda_i(x_i)\,\mathrm{d}x_i.$$

Using the tensor train (TT) format, we can efficiently approximate a d-dimensional function $\pi(x)$ and compute marginal probability distributions at low computational cost. To do so, we first define a d-dimensional discrete univariate grid over the parameter space $\mathcal{X}$, with n grid points in each dimension. In the tensor train format we can represent the function over this d-dimensional grid as a product train of 2D matrices (rank-2 tensor) and 3D matrices (rank-3 tensors), which we call TT-cores, see Fig. 2.4. More specifically each core $\pi_k \in \mathbb{R}^{r_{k-1} \times n \times r_k}$ has ranks r_{k-1} and r_k, for $k = 1, \ldots, d$, connecting it with its neighbouring cores, as illustrated in Figure 2.4. For the first and last cores, the outer ranks are set to $r_0 = r_d = 1$. This enables us to write the value $\pi(x)$, for a fixed point $x = (x_1, \ldots, x_d)$ on the grid, as a sequence of matrix multiplications

$$\pi_1(x_1)\pi_2(x_2) \cdots \pi_d(x_d) = \pi(x) \in \mathbb{R},$$

where each core $\pi_k(x_k)$, becomes a matrix of size $r_{k-1} \times r_k$. Clearly this shows that we only need dnr^2 evaluation points instead of n^d grid points to approximate the whole parameter space. Consequently, with a tensor train approximation, the marginal target function

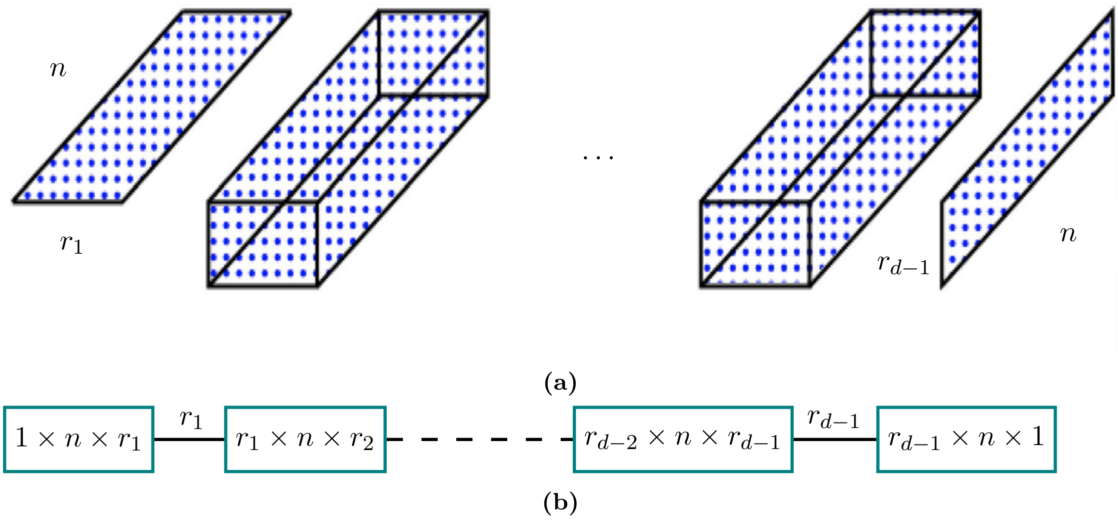

Figure 2.4: Here, we visualise the tensor train cores as two- and three-dimensional matrices. Each core has a length n, corresponding to the number of grid points in one dimension, and the cores are connected through ranks r_k. More specifically, a core π_k has dimensions $r_{k-1} \times n \times r_k$, with outer ranks $r_0 = r_d = 1$. Using the TT-format enables us to represent a d-dimensional grid with only dnr^2 evaluation points instead of n^d grid points. Figure (a) is adapted from [25].

$$
\begin{aligned}
f_{X_k}(x_k) = \frac{1}{z}\Big| & \left(\int_{\mathbb{R}} \lambda_1(x_1)\pi_1(x_1)\,\mathrm{d}x_1\right)\cdots\left(\int_{\mathbb{R}} \lambda_{k-1}(x_{k-1})\pi_{k-1}(x_{k-1})\,\mathrm{d}x_{k-1}\right) \\
& \lambda_k(x_k)\pi_k(x_k) \\
& \left(\int_{\mathbb{R}} \lambda_{k+1}(x_{k+1})\pi_{k+1}(x_{k+1})\,\mathrm{d}x_{k+1}\right)\cdots\left(\int_{\mathbb{R}} \lambda_d(x_d)\pi_d(x_d)\,\mathrm{d}x_d\right)\Big|
\end{aligned} \tag{2.33}
$$

is computed by integrating over all TT cores except π_k, as in [26], including a normalisation constant z [23].

In practice, tensor train approximations may suffer from numerical instability, particularly because it is not advantageous to approximate the target function $\pi(x)$ in for example, the logarithmic space. To address this, we follow the notation and procedure of Cui et al. [23] and instead approximate the square root of the probability density

$$\sqrt{\pi(x)} \approx g(x) = \boldsymbol{G}_1(x_1), \ldots, \boldsymbol{G}_k(x_k), \ldots, \boldsymbol{G}_d(x_d). \tag{2.34}$$

Here, each TT-core is given by

$$G_k^{(\alpha_{k-1},\alpha_k)}(x_k) = \sum_{i=1}^{n_k} \phi_k^{(i)}(x_k)\boldsymbol{A}_k[\alpha_{k-1}, i, \alpha_k], \quad k = 1, \ldots, d, \tag{2.35}$$

where $\boldsymbol{A}_k \in \mathbb{R}^{r_{k-1}\times n_k \times r_k}$ is the k-th coefficient tensor and $\{\phi_k^{(i)}(x_k)\}_{i=1}^{n_k}$ are the basis functions corresponding to the k-th coordinate. The approximated density is written as:

$$\pi(x) \approx \gamma' + g^2(x), \tag{2.36}$$

where γ' is a small positive constant added to ensure positivity and is chosen such that

$$\gamma' \leq \frac{1}{\boldsymbol{\lambda}(\mathcal{X})} \|g - \sqrt{\pi}\|_2^2. \tag{2.37}$$

This leads to the normalised target function

$$f_X(x) = \frac{1}{z}\boldsymbol{\lambda}(x)\pi(x) = \frac{1}{z}\left(\boldsymbol{\lambda}(x)\gamma' + \boldsymbol{\lambda}(x)g^2(x)\right), \tag{2.38}$$

where z is the normalisation constant. Given the tensor train approximation of $\sqrt{\pi}$, the marginal function $f_{X_k}(x_k)$ can be expressed as

$$\begin{aligned} f_{X_k}(x_k) = \frac{1}{z}\Bigg(&\gamma' \prod_{i=1}^{k-1} \boldsymbol{\lambda}_i(\mathcal{X}_i) \prod_{i=k+1}^{d} \boldsymbol{\lambda}_i(\mathcal{X}_i) \\ &+ \left(\int_{\mathbb{R}} \boldsymbol{\lambda}_1(x_1)\boldsymbol{G}_1^2(x_1)\,\mathrm{d}x_1\right) \cdots \left(\int_{\mathbb{R}} \boldsymbol{\lambda}_{k-1}(x_{k-1})\boldsymbol{G}_{k-1}^2(x_{k-1})\,\mathrm{d}x_{k-1}\right) \\ &\boldsymbol{\lambda}_k(x_k)\boldsymbol{G}_k^2(x_k) \\ &\left(\int_{\mathbb{R}} \boldsymbol{\lambda}_{k+1}(x_{k+1})\boldsymbol{G}_{k+1}^2(x_{k+1})\,\mathrm{d}x_{k+1}\right) \cdots \left(\int_{\mathbb{R}} \boldsymbol{\lambda}_d(x_d)\boldsymbol{G}_d^2(x_d)\,\mathrm{d}x_d\right)\Bigg). \end{aligned} \tag{2.39}$$

To compute these marginals efficiently, one can use a procedure similar to left and right orthogonalisation of TT-cores [27]. For this, we define the mass matrix $\boldsymbol{M}_k \in \mathbb{R}^{n_k \times n_k}$ as

$$\boldsymbol{M}_k[i,j] = \int_{\mathcal{X}_k} \phi_k^{(i)}(x_k)\phi_k^{(j)}(x_k)\boldsymbol{\lambda}(x_k)\,\mathrm{d}x_k, \quad i,j = 1,\ldots,n_k, \tag{2.40}$$

where $\{\phi_k^{(i)}(x_k)\}_{i=1}^{n_k}$ denotes the set of basis functions for the k-th coordinate.

2.6.1 Marginal Functions

We compute the marginal functions using two procedures, referred to as backward marginalisation [23] and forward marginalisation. The backward marginalisation provides us with the coefficient matrices $\boldsymbol{B}_k$, while the forward marginalisation gives the coefficient matrices $\boldsymbol{B}_{\text{pre},n}$. These matrices enable the efficient evaluation of marginal functions, similar to [23]. The proposition used to compute $\boldsymbol{B}_k$, stated in Proposition 1, is adapted directly from [23].

Proposition 1 (Backward Marginalisation)**:** Starting with the last coordinate $k = d$, we set $\boldsymbol{B}_d = \boldsymbol{A}_d$. The following procedure can be used to obtain the coefficient tensor $\boldsymbol{B}_{k-1} \in \mathbb{R}^{r_{k-2} \times n_{k-1} \times r_{k-1}}$, which we need for defining the marginal function $f_{X_k}(x_k)$:

1. Use the Cholesky decomposition of the mass matrix, $\boldsymbol{L}_k \boldsymbol{L}_k^\top = \boldsymbol{M}_k \in \mathbb{R}^{n_k \times n_k}$, to construct a tensor $\boldsymbol{C}_k \in \mathbb{R}^{r_{k-1} \times n_k \times r_k}$:

$$\boldsymbol{C}_k[\alpha_{k-1}, \tau, l_k] = \sum_{i=1}^{n_k} \boldsymbol{B}_k[\alpha_{k-1}, i, l_k] \boldsymbol{L}_k[i, \tau]. \tag{2.41}$$

2. Unfold $\boldsymbol{C}_k$ along the first coordinate and compute the thin QR decomposition, so that $\boldsymbol{C}_k^{(R)} \in \mathbb{R}^{r_{k-1} \times (n_k r_k)}$:

$$\boldsymbol{Q}_k \boldsymbol{R}_k = (\boldsymbol{C}_k^{(R)})^\top. \tag{2.42}$$

3. Compute the new coefficient tensor:

$$\boldsymbol{B}_{k-1}[\alpha_{k-2}, i, l_{k-1}] = \sum_{\alpha_{k-1}=1}^{r_{k-1}} \boldsymbol{A}_{k-1}[\alpha_{k-2}, i, \alpha_{k-1}] \boldsymbol{R}_k[l_{k-1}, \alpha_{k-1}]. \tag{2.43}$$

Proposition 2 (Forward Marginalisation)**:** Starting with the first coordinate $k = 1$, we set $\boldsymbol{B}_{\text{pre},1} = \boldsymbol{A}_1$. The following procedure can be used to obtain the coefficient tensor $\boldsymbol{B}_{\text{pre},k+1} \in \mathbb{R}^{r_k \times n_{k+1} \times r_{k+1}}$ for defining the marginal function $f_{X_k}(x_k)$:

1. Use the Cholesky decomposition of the mass matrix, $\boldsymbol{L}_k \boldsymbol{L}_k^\top = \boldsymbol{M}_k \in \mathbb{R}^{n_k \times n_k}$, to construct a tensor $\boldsymbol{C}_k \in \mathbb{R}^{r_{k-1} \times n_k \times r_k}$:

$$\boldsymbol{C}_{\text{pre},k}[\alpha_{k-1}, \tau, l_k] = \sum_{i=1}^{n_k} \boldsymbol{L}_k[i, \tau] \boldsymbol{B}_{\text{pre},k}[\alpha_{k-1}, i, l_k]. \tag{2.44}$$

2. Unfold $\boldsymbol{C}_{pre,k}$ along the first coordinate and compute the thin QR decomposition, so that $\boldsymbol{C}_{\text{pre},k}^{(R)} \in \mathbb{R}^{(r_{k-1} n_k) \times r_k}$:

$$\boldsymbol{Q}_{pre,k} \boldsymbol{R}_{\text{pre},k} = (\boldsymbol{C}_{\text{pre},k}^{(R)}). \tag{2.45}$$

3. Compute the new coefficient tensor $\boldsymbol{B}_{\text{pre},k+1} \in \mathbb{R}^{r_{k-1} \times n_k \times r_k}$:

$$\boldsymbol{B}_{\text{pre},k+1}[l_{k+1}, i, \alpha_{k+1}] = \sum_{\alpha_k=1}^{r_k} \boldsymbol{R}_{\text{pre},k}[l_{k+1}, \alpha_k] \boldsymbol{A}_{k+1}[\alpha_k, i, \alpha_{k+1}]. \tag{2.46}$$

After computing the coefficient tensors $\boldsymbol{B}_{\text{pre},k+1}$ as in Prop. 2 and $\boldsymbol{B}_{k+1}$ from Prop. 1, the marginal PDF of k-th dimension can be expressed as

$$f_{X_k}(x_k) = \frac{1}{z} \left(\gamma' \prod_{i=1}^{k-1} \lambda_i(X_i) \prod_{i=k+1}^{d} \lambda_i(X_i) + \sum_{l_{k-1}=1}^{r_{k-1}} \sum_{l_k=1}^{r_k} \left(\sum_{i=1}^{n} \phi_k^{(i)}(x_k) \boldsymbol{D}_k[l_{k-1}, i, l_k] \right)^2 \right) \lambda_k(x_k), \tag{2.47}$$

where $\boldsymbol{D}_k \in \mathbb{R}^{r_{k-1}\times n\times r_k}$ and $\boldsymbol{R}_{\text{pre},k-1} \in \mathbb{R}^{r_{k-1}\times r_{k-1}}$ and $\boldsymbol{B}_k \in \mathbb{R}^{r_{k-1}\times n\times r_k}$

$$\boldsymbol{D}_k[l_{k-1}, i, l_k] = \sum_{\alpha_{k-1}=1}^{r_{k-1}} \boldsymbol{R}_{\text{pre},k-1}[l_{k-1}, \alpha_{k-1}]\boldsymbol{B}_k[\alpha_{k-1}, i, l_k]. \tag{2.48}$$

For the first dimension, $f_{X_1}(x_1)$ can be expressed as

$$f_{X_1}(x_1) = \frac{1}{z}\left(\gamma' \prod_{i=2}^{d} \boldsymbol{\lambda}_i(\mathcal{X}_i) + \sum_{l_1=1}^{r_1}\left(\sum_{i=1}^{n}\phi_1^{(i)}(x_1)\boldsymbol{D}_1[i, l_1]\right)^2\right)\boldsymbol{\lambda}_1(x_1), \tag{2.49}$$

where $\boldsymbol{D}_1[i, l_1] = \boldsymbol{B}_1[\alpha_0, i, l_1]$ and $\alpha_0 = 1$, and similarly in the last dimension

$$f_{X_d}(x_d) = \frac{1}{z}\left(\gamma' \prod_{i=1}^{d-1} \boldsymbol{\lambda}_i(\mathcal{X}_i) + \sum_{l_{n-1}=1}^{r_{d-1}}\left(\sum_{i=1}^{n}\phi_1^{(i)}(x_1)\boldsymbol{D}_d[l_{n-1}, i]\right)^2\right)\boldsymbol{\lambda}_d(x_d), \tag{2.50}$$

where $\boldsymbol{D}_d[l_{n-1}, i] = \boldsymbol{B}_{\text{pre},d}[l_{n-1}, i, \alpha_{n+1}]$ and $\alpha_{d+1} = 1$.

[19] J. Andrés Christen and Colin Fox. "A general purpose sampling algorithm for continuous distributions (the t-walk)". In: *Bayesian Analysis* 5.2 (2010), pp. 263 –281.

[20] M. Capiński and P.E. Kopp. *Measure, Integral and Probability. Springer Undergraduate Mathematics Series.* London: Springer-Verlag London, 2004.

[21] M. Simonnet. *Measures and Probabilities.* New York: Springer-Verlag, 1996.

[22] Vesa Kaarnioja. *Inverse Problems. Eighth lecture.* `https://vesak90.userpage.fu-berlin.de/ip23/week8.pdf`. [Online; accessed 10/04/25]. 2023.

[23] Tiangang Cui and Sergey Dolgov. "Deep composition of tensor-trains using squared inverse rosenblatt transports". In: *Foundations of Computational Mathematics* 22.6 (2022), pp. 1863–1922.

[24] Philip J Davis and Philip Rabinowitz. *Methods of numerical integration.* San Diego, CA: Academic Press, Inc., 1984.

[25] Colin Fox et al. "Grid methods for Bayes-optimal continuous-discrete filtering and utilizing a functional tensor train representation". In: *Inverse Problems in Science and Engineering* 29.8 (2021), pp. 1199–1217.

[26] Sergey Dolgov et al. "Approximation and sampling of multivariate probability distributions in the tensor train decomposition". In: *Statistics and Computing* 30 (2020), pp. 603–625.

[27] Ivan V Oseledets. "Tensor-train decomposition". In: *SIAM Journal on Scientific Computing* 33.5 (2011), pp. 2295–2317.

[28] Australian National Concurrent Design Facility. *CubeSat Microwave Radiometer Mission to Support Global Ozone Layer Monitoring. Concept Study - Summary Report.* unpublished, internal report. Canberra BC: UNSW Canberra Space, 2023.

[29] Schwartz M. et al. *MLS/Aura Level 2 Ozone (O3) Mixing Ratio V005.* `https://disc.gsfc.nasa.gov/datasets/ML2O3_005/summary?keywords=mls%20o3`. [Online; accessed 25/04/24]. 2020.

[30] Marie Šimečková et al. "Einstein A-coefficients and statistical weights for molecular absorption transitions in the HITRAN database". In: *Journal of Quantitative Spectroscopy and Radiative Transfer* 98.1 (2006), pp. 130–155.

[31] J. Andrés Christen and Colin Fox. *The t-walk software.* `https://www.cimat.mx/~jac/twalk/`. [Online; accessed 25/11/24].

[32] Colin Fox. "Sampling and modelling of LARGE Gaussian processes". 9th World Meeting of International Society for Bayesian Analysis, 21st to 25th July 2008. Hamilton Island, Queensland, Australia. 2008. URL: `urllinktotalkabstractifany`.

[33] *U.S. Standard Atmosphere, 1976.* Washington, D.C.: United States. National Oceanic and Atmospheric Administration, United States Committee on Extension to the Standard Atmosphere, 1976.

[34] Johnathan M Bardsley et al. "Randomize-then-optimize for sampling and uncertainty quantification in electrical impedance tomography". In: *SIAM/ASA Journal on Uncertainty Quantification* 3.1 (2015), pp. 1136–1158.

[35] Josef Dick, Frances Y. Kuo, and Ian H. Sloan. "High-dimensional integration: The quasi-Monte Carlo way". In: *Acta Numerica* 22 (2013), 133–288.

[36] Ulli Wolff. *UWerr.m Version6.* `https://www.physik.hu-berlin.de/de/com/ALPHAsoft`. [Online; accessed 5/11/23]. 2004.

[37] Greg Lawler. *Notes on probability.* `https://www.math.uchicago.edu/~lawler/probnotes.pdf`. [Online; accessed 10/04/25]. 2016.