Available online at www.sciencedirect.com

ScienceDirect

Journal of Approximation Theory 241 (2019) 33–56

JOURNAL OF Approximation Theory

www.elsevier.com/locate/jat

Full Length Article

The difficulty of Monte Carlo approximation of multivariate monotone functions

Robert J. Kunsch

Universität Osnabrück, Institut für Mathematik, Albrechtstr. 28a, 49076 Osnabrück, Germany

Received 15 March 2018; received in revised form 20 November 2018; accepted 8 January 2019
Available online 15 January 2019

Communicated by Josef Dick

Abstract

We study the L_1-approximation of d-variate monotone functions based on information from n function evaluations. It is known that this problem suffers from the curse of dimensionality in the deterministic setting, that is, the number $n(\varepsilon, d)$ of function evaluations needed in order to approximate an unknown monotone function within a given error threshold ε grows at least exponentially in d. In the randomized setting (Monte Carlo setting) the complexity $n(\varepsilon, d)$ grows exponentially in $\sqrt{d}$ (modulo logarithmic terms) only. An algorithm exhibiting this complexity is presented. The problem remains difficult as best methods known are deterministic if ε is comparably small, namely $\varepsilon \preceq 1/\sqrt{d}$. This inherent difficulty is confirmed by lower complexity bounds which reveal a joint (ε, d)-dependence and from which we deduce that the problem is *not weakly tractable*. The lower bound proof also has implications on the complexity of learning Boolean monotone functions.

Keywords: Monte Carlo approximation; Monotone functions; Information-based complexity; Standard information; Intractable; Curse of dimensionality

1. Introduction

In this paper we consider the L_1-approximation of d-variate *monotone* functions using function values as information,

$$\mathrm{APP} : F_{\mathrm{mon}}^d \hookrightarrow L_1([0,1]^d), \quad f \mapsto f,$$

E-mail address: robert.kunsch@uni-osnabrueck.de.

https://doi.org/10.1016/j.jat.2019.01.003
0021-9045/

where the input set

$$F^d_{\mathrm{mon}} := \{f : [0,1]^d \to [-1,1] \mid \mathbf{x} \le \widetilde{\mathbf{x}} \Rightarrow f(\mathbf{x}) \le f(\widetilde{\mathbf{x}})\}$$

consists of monotonically increasing functions with respect to the partial order on the domain. For $\mathbf{x}, \widetilde{\mathbf{x}} \in \mathbb{R}^d$, the partial order is defined by

$$\mathbf{x} \le \widetilde{\mathbf{x}} \qquad :\Leftrightarrow \qquad x_j \le \widetilde{x}_j \text{ for all } j = 1, \dots, d\,.$$

The approximation of monotone functions is not a linear problem according to the book *Information-based Complexity* (IBC) by Traub et al. [20] because the set F^d_{mon} is not symmetric: For non-constant functions $f \in F^d_{\mathrm{mon}}$, the negative $-f$ is not contained in F^d_{mon} since it will be monotonically decreasing. The monotonicity assumption is different from common smoothness assumptions, yet it implies many other nice properties, see for example Alberti and Ambrosio [1]. Integration and approximation of monotone functions have been studied in several papers [10,14,17] to which we will refer in the course of this paper. Monotonicity can also be an assumption for statistical problems [7,18]. A similar structural assumption could be convexity (more general: k-monotonicity), numerical problems with such properties have been studied for example in [6,10–12,15].

In approximating monotone functions with respect to the L_1-norm, a deterministic algorithm with *cardinality* n is a mapping

$$A_n : F^d_{\mathrm{mon}} \xrightarrow{N} \mathbb{R}^n \xrightarrow{\phi} L_1([0,1]^d)\,,$$

where N is the *information mapping*

$$N(f) = (y_1, \dots, y_n) := (f(\mathbf{x}_1), \dots, f(\mathbf{x}_n))\,.$$

The nodes $\mathbf{x}_1, \dots, \mathbf{x}_n$ may be selected in an adaptive manner, that is, the choice of the node $\mathbf{x}_i$ may depend on previously obtained information $y_1, \dots, y_{i-1}$. (One could even adaptively vary the number n of computed function values, thereby building an algorithm with so-called *varying cardinality*.) The *worst case error* of such a method is defined by

$$e(A_n, F^d_{\mathrm{mon}}) := \sup_{f \in F^d_{\mathrm{mon}}} \|A_n(f) - f\|_{L_1}\,.$$

A *Monte Carlo method* $A_n = (A^\omega_n)_{\omega \in \Omega}$ is a family of such mappings indexed by a random element ω from a probability space $(\Omega, \Sigma, \mathbb{P})$. Hence, fixing an input f, the output $A_n(f)$ is a random variable with values in $L_1([0,1]^d)$. We assume that the *Monte Carlo error*,

$$e((A^\omega_n), F^d_{\mathrm{mon}}) := \sup_{f \in F^d_{\mathrm{mon}}} \mathbb{E}\|A_n(f) - f\|_{L_1}\,,$$

is well defined, in particular, the error functional $\omega \mapsto \|A^\omega_n(f) - f\|_{L_1}$ shall be measurable for any input f. Our goal is to compare the deterministic setting (worst case) with the randomized setting (Monte Carlo) in terms of the respective minimal errors using information of cardinality $n \in \mathbb{N}_0$,

$$e^{\mathrm{det}}(n, F^d_{\mathrm{mon}}) := \inf_{A_n} e(A_n, F^d_{\mathrm{mon}}) \qquad \text{vs.} \qquad e^{\mathrm{ran}}(n, F^d_{\mathrm{mon}}) := \inf_{(A^\omega_n)} e((A^\omega_n), F^d_{\mathrm{mon}})\,,$$

and also in terms of the respective complexities for accuracy $\varepsilon > 0$ and dimension d,

$$n^{\mathrm{det}}(\varepsilon, F^d_{\mathrm{mon}}) := \inf\{n \in \mathbb{N}_0 \mid \exists A_n : e(A_n, F^d_{\mathrm{mon}}) \le \varepsilon\}$$
$$\text{vs.} \qquad n^{\mathrm{ran}}(\varepsilon, F^d_{\mathrm{mon}}) := \inf\{n \in \mathbb{N}_0 \mid \exists (A^\omega_n) : e((A^\omega_n), F^d_{\mathrm{mon}}) \le \varepsilon\}\,.$$

In Section 2 we present a collection of results that are known from previous papers or can be shown with well known techniques. In Section 2.1 we show that for fixed dimension d the order of convergence for the L_1-approximation of monotone functions cannot be improved by randomization. In Section 2.2 we cite Hinrichs, Novak and Woźniakowski [10], who showed that the deterministic complexity grows exponentially in d for some fixed ε, this is called the *curse of dimensionality*, a notion coined by Bellman 1957 [3]. Monte Carlo methods may "mitigate" the curse by significantly reducing the d-dependence of the complexity, as long as the accuracy ε is fixed. Namely, in Section 3 we show the combined upper bound

$$n^{\mathrm{ran}}(\varepsilon, F^d_{\mathrm{mon}}) \;\le\; \min\left\{\exp\left[C\,\frac{\sqrt{d}}{\varepsilon}\,\left(1+\log\frac{d}{\varepsilon}\right)^{3/2}\right],\; \exp\left[d\,\log\frac{d}{2\varepsilon}\right]\right\}, \tag{1}$$

with some numerical constant $C > 0$, see Theorem 3.3. Here, the first bound is achieved by a proper Monte Carlo method and applies in the pre-asymptotic regime. In particular, for fixed ε the complexity grows exponentially in $\sqrt{d}$ (modulo logarithmic terms) only. Despite this speed-up by an exponent $\sqrt{d}$, and although formally the curse does not hold anymore – according to the definition commonly used in IBC [9,10,16,21] – the Monte Carlo complexity still grows very fast with the dimension. Even worse, the second bound in (1) is achieved by deterministic algorithms and applies for small error thresholds $\varepsilon \preceq 1/\sqrt{d}$ (modulo logarithmic terms). Lower bounds for the Monte Carlo setting are found in Section 4, where we prove

$$n^{\mathrm{ran}}(\varepsilon, F^d_{\mathrm{mon}}) \;>\; \nu\,\exp\left[c\,\frac{\sqrt{d}}{\varepsilon}\right], \qquad \text{for } \varepsilon_0\sqrt{d_0/d} \le \varepsilon \le \varepsilon_0 \text{ and } d \ge d_0, \tag{2}$$

with numerical constants $\nu, c, \varepsilon_0 > 0$ and $d_0 \in \mathbb{N}$, see Theorem 4.1. There is a constraint on ε, which is not surprising as it fits to the observation that for smaller ε best algorithms known are deterministic and we have a different joint (ε, d)-dependence in that regime. However, by monotonicity of the ε-complexity, we can still conclude

$$n^{\mathrm{ran}}(\varepsilon, F^d_{\mathrm{mon}}) \;>\; \nu\,\exp\left[c'\,d\right], \qquad \text{for } 0 < \varepsilon \le \varepsilon_0\sqrt{d_0/d} \text{ and } d \ge d_0, \tag{3}$$

where $c' = c/(\varepsilon_0\sqrt{d_0})$. Hence, the lower bounds match the upper bounds except for logarithmic terms in the exponent. For moderately decaying error thresholds $\varepsilon = \varepsilon_0\sqrt{d_0/d}$, the Monte Carlo complexity depends already exponentially on d, we conclude that the problem is *not weakly tractable*, we thus call it *intractable*, see Remark 4.2.

This paper is concerned with real-valued monotone functions $f : [0,1]^d \to [-1,1]$. A closely related problem is the approximation of *Boolean* monotone functions $f : \{0,1\}^d \to \{0,1\}$. The algorithm we present in Section 3 is inspired by method for learning Boolean monotone functions due to Bshouty and Tamon [5]. The Monte Carlo lower bounds given in Section 4 are actually obtained by a reduction to the approximation of Boolean monotone functions. It is then a modification of a lower bound proof which can be found in Blum, Burch and Langford [4]. Similarly to the real-valued setting in Section 2.2, one can show the curse of dimensionality for deterministic approximation of Boolean monotone functions as well, see the author's PhD thesis [13, Theorem 4.5]. The main difference to real-valued monotone functions is that the concept of order of convergence, see Section 2.1, is meaningless for a discrete problem such as the approximation of Boolean functions.

2. Survey on deterministic approximation

2.1. The classical approach—order of convergence

The classical approach for the numerical analysis of multivariate problems is to fix the dimension d and to study the order at which the error $e(n)$ converges to zero as the information budget n grows. We will use the common asymptotic notation

$$a_n \preceq e(n) \preceq b_n \qquad :\Leftrightarrow \qquad \exists\, c, C > 0\colon\ c\, a_n \le e(n) \le C\, b_n\,.$$

If $a_n \preceq e(n) \preceq a_n$, we simply write $e(n) \asymp a_n$. The hidden prefactors c and C may depend on problem parameters such as the dimension. Sometimes, while knowing the exact order $e(n) \asymp a_n$, we still have a gap between the prefactors that grows unpleasantly with the dimension.

As an example, the order of convergence has been studied for approximating the integral of monotone functions,

$$\mathrm{INT} : F^d_{\mathrm{mon}} \to \mathbb{R}, \quad f \mapsto \int_{[0,1]^d} f \,\mathrm{d}\mathbf{x}\,,$$

based on finitely many function evaluations. Interestingly, for this problem adaption makes a difference in the randomized setting (at least for $d = 1$), but non-adaptive randomization helps only for $d \ge 2$ to speed up the convergence compared to deterministic methods. In the univariate case Novak [14] showed

$$\begin{aligned} &e^{\mathrm{ran,ada}}(n, \mathrm{INT}, F^1_{\mathrm{mon}}) \asymp n^{-3/2} \\ &\quad \prec e^{\mathrm{ran,nonada}}(n, \mathrm{INT}, F^1_{\mathrm{mon}}) \asymp e^{\mathrm{det}}(n, \mathrm{INT}, F^1_{\mathrm{mon}}) \asymp n^{-1}\,. \end{aligned}$$

Papageorgiou [17] examined the integration of d-variate monotone functions, for dimensions $d \ge 2$ we have

$$\begin{aligned} &e^{\mathrm{ran,ada}}(n, \mathrm{INT}, F^d_{\mathrm{mon}}) \asymp n^{-1/d-1/2} \\ &\quad \preceq e^{\mathrm{ran,nonada}}(n, \mathrm{INT}, F^d_{\mathrm{mon}}) \preceq n^{-1/(2d)-1/2} \\ &\quad \prec e^{\mathrm{det}}(n, \mathrm{INT}, F^d_{\mathrm{mon}}) \asymp n^{-1/d}\,, \end{aligned}$$

where the hidden prefactors depend on d. It is an open problem to find lower bounds for the non-adaptive Monte Carlo error that actually show that adaption is better for $d \ge 2$ as well, but from the one-dimensional case we conjecture it to be like that.

For the L_1-approximation of monotone functions, however, the order of convergence does not reveal any differences between the various algorithmic settings. Applying Papageorgiou's proof technique to this problem, we obtain the following theorem.

Theorem 2.1. *For the L_1-approximation of monotone functions, for fixed dimension d and $n \to \infty$, we have the following asymptotic behaviour,*

$$e^{\mathrm{ran}}(n, \mathrm{APP}, F^d_{\mathrm{mon}}) \asymp e^{\mathrm{det}}(n, \mathrm{APP}, F^d_{\mathrm{mon}}) \asymp n^{-1/d}\,,$$

where the implicit prefactors depend on d.

Proof. We split $[0,1]^d$ into m^d subcubes indexed by $\mathbf{i} \in \{0, 1, \ldots, m-1\}^d$:

$$C_{\mathbf{i}} := \bigtimes_{j=1}^{d} I_{i_j}$$

where $I_i := [\frac{i}{m}, \frac{i+1}{m})$ for $i = 0, 1, \ldots, m-2$, and $I_{m-1} := [\frac{m-1}{m}, 1]$.

For the lower bounds, we consider fooling functions $f = f_\delta$ that are constant on each of the subcubes, in detail,

$$f|_{C_\mathbf{i}} = \frac{2(|\mathbf{i}|_1 + \delta_\mathbf{i})}{d(m-1)+1} - 1$$

with $\delta_\mathbf{i} \in \{0, 1\}$ and $|\mathbf{i}|_1 := i_1 + \ldots + i_d$. Obviously, such functions are monotonically increasing. In order to obtain lower bounds that hold for Monte Carlo algorithms, we employ a minimax argument, also known as *Bakhvalov's trick* [2]. Namely, we average over all possible settings of $\boldsymbol{\delta} = (\delta_\mathbf{i})$. For any information $\mathbf{y}$, let $I^\mathbf{y} \subset \{0, \ldots, m-1\}^d$ be the set of indices $\mathbf{i}$ where we do not know anything about the function on the corresponding subcube $C_\mathbf{i}$. For an algorithm which uses $n < m^d$ function values, we have $\#I^\mathbf{y} \geq m^d - n$. Considering an arbitrary Monte Carlo algorithm $A_n^\omega = \phi^\omega \circ N^\omega$, we can write

$$\begin{aligned} e((A_n^\omega), F_{\mathrm{mon}}^d) &= \sup_{f \in F_{\mathrm{mon}}^d} \mathbb{E}\|A_n^\omega(f) - f\|_{L_1} \geq 2^{-m^d} \sum_{\delta \in \{0,1\}^{m^d}} \mathbb{E}\|A_n^\omega(f_\delta) - f_\delta\|_{L_1} \\ &\geq \mathbb{E} 2^{-m^d} \sum_{\delta \in \{0,1\}^{m^d}} \sum_{\substack{\mathbf{i} \in I^\mathbf{y} \\ \text{where } \mathbf{y} := N^\omega(f_\delta) \\ \text{and } g := \phi^\omega(\mathbf{y})}} \int_{C_\mathbf{i}} \underbrace{\frac{1}{2}\left(\left| \frac{2|\mathbf{i}|_1}{d(m-1)+1} - g(\mathbf{x}) \right| + \left| \frac{2(|\mathbf{i}|_1+1)}{d(m-1)+1} - g(\mathbf{x}) \right| \right)}_{\geq \frac{1}{d(m-1)+1}} \mathrm{d}\mathbf{x} \\ &\geq \left(1 - \frac{n}{m^d}\right) \frac{1}{d(m-1)+1} \,. \end{aligned}$$

Choosing $m := \lceil (2n)^{1/d} \rceil$, we obtain the general lower bound

$$e^{\mathrm{det}}(n, \mathrm{APP}, F_{\mathrm{mon}}^d) \geq e^{\mathrm{ran}}(n, \mathrm{APP}, F_{\mathrm{mon}}^d) \geq \frac{1}{2\,(d \cdot (2n)^{1/d} + 1)} \geq \frac{1}{6\,d}\, n^{-1/d} \,.$$

For the upper bounds, we give a deterministic, non-adaptive algorithm with cardinality $(m-1)^d$, i.e. when allowed to use n function values, we choose $m := \lfloor n^{1/d} \rfloor + 1$. Splitting the domain into m^d subcubes as above, we compute $(m-1)^d$ function values at the corner points in the interior $(0, 1)^d$ of the domain. For each subcube we take the arithmetic mean of the functions values of f in the lower and the upper corner of that particular subcube, where at the boundary without computing function values we assume

$$f|_{[0,1)^d \setminus (0,1)^d} = -1 \quad \text{and} \quad f|_{[0,1]^d \setminus [0,1)^d} = 1 \,.$$

The subcubes can be grouped along diagonals, where the upper corner of one subcube touches the lower corner of the next subcube. Each diagonal can be uniquely represented by an index $\mathbf{i}$ with at least one 0-entry, which thus belongs to the lowest subcube $C_\mathbf{i}$ of that diagonal, in total we have $m^d - (m-1)^d \leq d\, m^{d-1}$ diagonals. Due to monotonicity, the contribution of a single diagonal to the L_1-error is at most m^{-d}, so altogether we have

$$e(A_m^d, f) \;\leq\; \frac{d\, m^{d-1}}{m^d} = \frac{d}{m} = \frac{d}{\lfloor n^{1/d} \rfloor + 1} \;\leq\; d\, n^{-1/d} \,.$$

For more details see the proofs for integration in Papageorgiou [17]. □

The Monte Carlo lower bound contained in the above result also holds for algorithms with varying cardinality, see [13, Theorem 4.2].

Remark 2.2. The above proof yields the explicit estimate

$$\frac{1}{6\,d}\, n^{-1/d} \;\leq\; e^{\mathrm{ran}}(n, \mathrm{APP}, F_{\mathrm{mon}}^d) \;\leq\; e^{\mathrm{det}}(n, \mathrm{APP}, F_{\mathrm{mon}}^d) \;\leq\; d\, n^{-1/d} \,.$$

The ratio between the upper and the lower bound is only polynomial in d, but the picture is totally different when switching to the notion of ε-complexity for $0 < \varepsilon < 1$:

$$\left(\frac{1}{6\,d}\right)^{d} \varepsilon^{-d} \;\le\; n^{\mathrm{ran}}(\varepsilon, \mathrm{APP}, F^d_{\mathrm{mon}}) \;\le\; n^{\mathrm{det}}(\varepsilon, \mathrm{APP}, F^d_{\mathrm{mon}}) \;\le\; d^d\, \varepsilon^{-d}\,.$$

Here, the gap is super-exponential in d. In particular, the upper bound is based on algorithms that use super-exponentially (in d) many function values, which makes it essentially impossible to apply them in high dimensions.

It turns out that the cardinality of deterministic methods that achieve a fixed accuracy ε is always at least exponential in d, see Section 2.2. For the randomized setting, however, we can significantly reduce the d-dependence (which is still high), at least as long as ε is fixed, see Section 3. To summarize, if we only consider the order of convergence, we might think that randomization does not help, but for high dimensions randomization actually *does* help, at least in the *pre-asymptotic* regime.

2.2. Curse of dimensionality in the deterministic setting

Hinrichs, Novak, and Woźniakowski [10] have shown that the integration (and hence also the L_p-approximation, $1 \le p \le \infty$) of monotone functions suffers from the curse of dimensionality in the deterministic setting.

Theorem 2.3 (*Hinrichs, Novak, Woźniakowski 2011*)**.** *The L_1-approximation of monotone functions suffers from the curse of dimensionality in the deterministic setting. In particular,*

$$e^{\mathrm{det}}(n, F^d_{\mathrm{mon}}) \ge \left(1 - n\,2^{-d}\right)\,,$$

so for $0 < \varepsilon \le 1/2$ we have

$$n^{\mathrm{det}}(\varepsilon, F^d_{\mathrm{mon}}) \ge 2^{d-1}\,.$$

Proof (*Idea of the proof*)**.** Any deterministic algorithm will fail to distinguish the diagonal split function $f_{\diagdown}(\mathbf{x}) := \mathrm{sgn}\left(\sum_{j=1}^d x_j - \frac{d}{2}\right)$ from other monotone functions F^d_{mon} which yield the same information. No matter what information mapping N we take, there will exist such indistinguishable functions with a big L_1-distance to $f_{\diagdown}$, since in this situation each function value only provides knowledge about a subdomain of volume at most 2^{-d}, see [10] for details. □

Note that the *initial error* $e(0, F^d_{\mathrm{mon}})$ is 1, this means, if we do not know any function value, the best guess is the zero function. Thus the theorem above states that in order to merely halve the initial error we already need exponentially (in d) many function values. The curse of dimensionality can be "mitigated" via Monte Carlo, see Section 3, but we still have intractability in the randomized setting, see Section 4. In contrast, for *integration* the standard Monte Carlo method

$$M_n(f) := \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{X}_i) \;\approx\; \mathrm{INT}(f)\,, \qquad \text{where } \mathbf{X}_i \overset{\mathrm{iid}}{\sim} \mathrm{unif}([0,1]^d),$$

easily achieves *strong polynomial tractability*, namely $n(\varepsilon, \mathrm{INT}, F^d_{\mathrm{mon}}) \le \lceil \varepsilon^{-2} \rceil$, where the dimension d does not play any role, the curse is broken.

3. Pre-asymptotic speed-up by Monte Carlo

We present and analyse a new algorithm for the approximation of monotone functions on the unit cube. It is the first algorithm to show that Monte Carlo significantly reduces the dimension dependence of the complexity compared to deterministic methods. The idea of the algorithm has been inspired by a method for learning Boolean monotone functions due to Bshouty and Tamon [5].

The method is based on a decomposition of the function f into tensorized Haar wavelets. We define dyadic cuboids on $[0,1]^d$ indexed by $\boldsymbol{\alpha} \in \mathbb{N}^d$, or equivalently by an index vector pair $(\boldsymbol{\lambda}, \boldsymbol{\kappa})$ with $\boldsymbol{\lambda} \in \mathbb{N}_0^d$ and $\boldsymbol{\kappa} \in \mathbb{N}_0^d$, $\kappa_j < 2^{\lambda_j}$, such that $\alpha_j = 2^{\lambda_j} + \kappa_j$ for $j = 1, \dots, d$:

$$C_{\boldsymbol{\alpha}} = C_{\boldsymbol{\lambda},\boldsymbol{\kappa}} := \bigtimes_{j=1}^{d} I_{\alpha_j},$$

where

$$I_{\alpha_j} = I_{\lambda_j,\kappa_j} := \begin{cases} [\kappa_j\, 2^{-\lambda_j}, (\kappa_j+1)\, 2^{-\lambda_j}) & \text{for } \kappa_j = 0, \dots, 2^{\lambda_j} - 2, \\ [1 - 2^{-\lambda_j}, 1] & \text{for } \kappa_j = 2^{\lambda_j} - 1. \end{cases}$$

Note that for fixed λ_j we have a decomposition of the unit interval $[0,1]$ into 2^{λ_j} disjoint intervals of length $2^{-\lambda_j}$. (This index system for subdomains differs from the index system for subcubes in Section 2.1, which shall be no source of confusion.) One-dimensional Haar wavelets $\psi_{\alpha_j} : [0,1] \to \mathbb{R}$ are defined for $\alpha_j \in \mathbb{N}_0$ (if $\alpha_j = 0$, we put $\lambda_j = -\infty$ and $\kappa_j = 0$),

$$\psi_{\alpha_j} := \begin{cases} \mathbb{1}_{[0,1]} & \text{if } \alpha_j = 0 (\text{i.e. } \lambda_j = -\infty \text{ and } \kappa_j = 0), \\ 2^{\lambda_j/2} \left(\mathbb{1}_{I_{\lambda_j+1,2\kappa_j+1}} - \mathbb{1}_{I_{\lambda_j+1,2\kappa_j}}\right) & \text{if } \alpha_j \geq 1 (\text{i.e. } \lambda_j \geq 0). \end{cases}$$

In $L_2([0,1]^d)$ we have the orthonormal tensor product basis $\{\psi_{\boldsymbol{\alpha}}\}_{\boldsymbol{\alpha} \in \mathbb{N}_0^d}$ with

$$\psi_{\boldsymbol{\alpha}}(\mathbf{x}) := \prod_{j=1}^{d} \psi_{\alpha_j}(x_j).$$

The volume of the support of $\psi_{\boldsymbol{\alpha}}$ is $2^{-|\boldsymbol{\lambda}|_+}$ with $|\boldsymbol{\lambda}|_+ := \sum_{j=1}^d \max\{0, \lambda_j\}$. The basis function $\psi_{\boldsymbol{\alpha}}$ only takes discrete values $\{0, \pm 2^{|\boldsymbol{\lambda}|_+/2}\}$, hence it is normalized indeed. (This basis is well suited for monotone functions but differs from the usual Haar basis, mainly as the multidimensional Haar basis is not simply the tensor product basis of the one-dimensional basis. Further, for convenience, we switched signs for the one-dimensional Haar wavelets ψ_α with $\alpha > 0$.)

We can write any monotone function f as the decomposition

$$f = \sum_{\boldsymbol{\alpha} \in \mathbb{N}_0^d} \widetilde{f}(\boldsymbol{\alpha})\, \psi_{\boldsymbol{\alpha}}$$

with the coefficients

$$\widetilde{f}(\boldsymbol{\alpha}) := \langle \psi_{\boldsymbol{\alpha}}, f \rangle = \mathbb{E} \psi_{\boldsymbol{\alpha}}(\mathbf{X})\, f(\mathbf{X}),$$

where $\mathbf{X}$ is uniformly distributed on $[0,1]^d$. The algorithm shall use information from random samples $f(\mathbf{X}_1), \dots, f(\mathbf{X}_n)$, with $\mathbf{X}_i \overset{\text{iid}}{\sim} \text{unif}[0,1]^d$, in order to approximate the most important coefficients via the standard Monte Carlo method,

$$\widetilde{f}(\boldsymbol{\alpha}) \approx \widetilde{h}(\boldsymbol{\alpha}) := \frac{1}{n} \sum_{i=1}^{n} \psi_{\boldsymbol{\alpha}}(\mathbf{X}_i)\, f(\mathbf{X}_i)\,. \tag{4}$$

In particular, we choose a resolution $r \in \mathbb{N}$, and a parameter $k \in \{1, \dots, d\}$, and only consider indices $\boldsymbol{\alpha} \leftrightarrow (\boldsymbol{\lambda}, \boldsymbol{\kappa})$ with $\lambda_j < r$ and $|\boldsymbol{\alpha}|_0 := \#\{j \mid \alpha_j > 0\} \le k$. (The quantity $|\boldsymbol{\alpha}|_0$ counts the number of *active variables* of a basis function $\psi_{\boldsymbol{\alpha}}$.) A naive linear algorithm would simply return a linear reconstruction,

$$h = A^{\omega}_{n,k,r}(f) := \sum_{\substack{\boldsymbol{\alpha} \in \mathbb{N}_0^d \\ |\boldsymbol{\alpha}|_0 \le k \\ \boldsymbol{\lambda} < r}} \widetilde{h}(\boldsymbol{\alpha})\, \psi_{\boldsymbol{\alpha}}\,, \qquad \text{for } f \in F^d_{\text{mon}}\,. \tag{5}$$

This linear algorithm already provides the correct d-dependence (exponential in $\sqrt{d}$ modulo logarithmic terms) but the ε-dependence of the required sample size is unfavourable, see [13, Theorem 4.22] for a detailed analysis. Instead, for the subclass of sign-valued monotone functions,

$$F^d_{\text{mon}\pm} := \{f : [0,1]^d \to \{-1, +1\} \mid f \in F^d_{\text{mon}}\}\,,$$

in the L_1-approximation setting it is natural to return a sign-valued approximation,

$$g = \widehat{A}^{\omega}_{n,k,r}(f) := \operatorname{sgn} h = \operatorname{sgn} \circ\, [A^{\omega}_{n,k,r}(f)]\,, \qquad \text{for } f \in F^d_{\text{mon}\pm}\,. \tag{6}$$

(Here and throughout the paper we put $\operatorname{sgn}(0) := 1$ in order to avoid zero values.) For general monotone functions $f \in F^d_{\text{mon}}$ with function values in $[-1, +1]$, the algorithm can be generalized to

$$\bar{A}^{\omega}_{n,k,r}(f) := \frac{1}{2} \int_{-1}^{1} \widehat{A}^{\omega}_{n,k,r}(f_t)\, \mathrm{d}t\,, \qquad \text{where } f_t(\mathbf{x}) := \operatorname{sgn}(f(\mathbf{x}) - t) \text{ for } t \in \mathbb{R}. \tag{7}$$

Note that the function values $f_t(\mathbf{X}_i)$ which are needed in the course of evaluating $\bar{A}_{n,k,r}$ can be directly derived from function values $f(\mathbf{X}_i)$, so we still use the same information as within the simple linear algorithm (5). (This trick would not be possible for algorithms with an adaptive procedure for collecting information on a sign-valued function.) The idea for the generalized algorithm $\bar{A}_{n,k,r}$ is based on the observation

$$f(\mathbf{x}) = \frac{1}{2} \int_{-1}^{1} \operatorname{sgn}(f(\mathbf{x}) - t)\, \mathrm{d}t\,, \qquad \text{for } f(\mathbf{x}) \in [-1, 1]. \tag{8}$$

The validity of this approach is summarized in the following Lemma.

Lemma 3.1. *For the approximation of monotone functions with the methods* $\widehat{A}_{n,k,r} = (\widehat{A}^{\omega}_{n,k,r})_{\omega \in \Omega}$ *defined in* (6) *and* $\bar{A}_{n,k,r} = (\bar{A}^{\omega}_{n,k,r})_{\omega \in \Omega}$ *from* (7) *we have*

$$e(\widehat{A}_{n,k,r}, F^d_{\text{mon}\pm}) = e(\bar{A}_{n,k,r}, F^d_{\text{mon}})\,.$$

Proof. Since for sign-valued functions $f \in F^d_{\text{mon}\pm}$ we have $f_t = f$ for $t \in (-1, 1]$, trivially $\widehat{A}_{n,k,r}(f) = \bar{A}_{n,k,r}(f)$, and from $F^d_{\text{mon}\pm} \subset F^d_{\text{mon}}$ we conclude the inequality $e(\widehat{A}_{n,k,r}, F^d_{\text{mon}\pm}) \le e(\bar{A}_{n,k,r}, F^d_{\text{mon}})$.

Conversely, for $f \in F^d_{\mathrm{mon}}$, using the definition of $\bar{A}_{n,k,r}$ in (7) and the observation (8), via the triangle-inequality and Fubini's theorem we have

$$\begin{aligned} e(\bar{A}_{n,k,r},\, f) &= \mathbb{E}\left\| f - \bar{A}^{\omega}_{n,k,r}(f)\right\|_{L_1} \\ [(7),(8)] \qquad &= \mathbb{E}\left\| \frac{1}{2}\int_{-1}^{1} (f_t - \widehat{A}^{\omega}_{n,k,r}(f_t))\,\mathrm{d}t \right\|_{L_1} \\ [\Delta\text{-ineq., Fubini}] \qquad &\le \frac{1}{2}\int_{-1}^{1} \mathbb{E}\left\| f_t - \widehat{A}^{\omega}_{n,k,r}(f_t)\right\| \mathrm{d}t \;\le\; e(\widehat{A}_{n,k,r},\, F^d_{\mathrm{mon}\pm}) \,. \end{aligned}$$

This implies $e(\bar{A}_{n,k,r},\, F^d_{\mathrm{mon}}) \le e(\widehat{A}_{n,k,r},\, F^d_{\mathrm{mon}\pm})$, thus finishing the proof. □

We continue with the error analysis of the given algorithm, where by virtue of the above lemma we may restrict to the approximation of sign-valued monotone functions via $\widehat{A}_{n,k,r}$. For details on the implementation of $\bar{A}_{n,k,r}$, see Remark 3.5.

A key result for the error analysis is the following fact about those coefficients which are dropped by the algorithm, compare to Bshouty and Tamon [5, Section 4] for the Boolean setting.

Lemma 3.2. *For any monotone function $f \in F^d_{\mathrm{mon}}$ we have*

$$\sum_{\substack{\boldsymbol{\alpha}\in\mathbb{N}_0^d \\ |\boldsymbol{\alpha}|_0 > k \\ \boldsymbol{\lambda} < r}} \widetilde{f}(\boldsymbol{\alpha})^2 \;\le\; \frac{\sqrt{d\,r}}{k+1} \,.$$

Proof. Within the first step, we consider special coefficients $\widetilde{f}(\alpha\,\mathbf{e}_j)$ with one active variable. These measure the average growth of f along the jth coordinate within the interval I_α, where $\mathbf{e}_j$ is the jth vector of the standard basis in $\mathbb{R}^d$. We will frequently use the alternative indexing $I_{\lambda,\kappa}$ with $\alpha = 2^\lambda + \kappa \in \mathbb{N}$, where $\lambda \in \mathbb{N}_0$ and $\kappa = 0, \dots, 2^\lambda - 1$. We define the function

$$f_{\alpha,j}(\mathbf{x}) := \begin{cases} 0 & \text{for } x_j \notin I_\alpha, \\ 2^{\lambda/2} \int_0^1 \psi_\alpha(z_j) \cdot f(\mathbf{z})\Big|_{\substack{z_{j'}=x_{j'} \\ \text{for } j'\neq j}} \mathrm{d}z_j & \text{for } x_j \in I_\alpha. \end{cases}$$

Due to the monotonicity of f we have $f_{\alpha,j} \ge 0$, and from the boundedness of f we have $f_{\alpha,j} \le 1$. Using this and Parseval's equation, we obtain

$$\widetilde{f}(\alpha\,\mathbf{e}_j) = \langle \psi_{\alpha\,\mathbf{e}_j},\, f\rangle = 2^{\lambda/2}\left\| f_{\alpha,j}\right\|_{L_1} \ge 2^{\lambda/2}\left\| f_{\alpha,j}\right\|^2_{L_2} = 2^{\lambda/2} \sum_{\boldsymbol{\alpha}'\in\mathbb{N}_0^d} \left\langle \psi_{\boldsymbol{\alpha}'},\, f_{\alpha,j}\right\rangle^2 .$$

Since the function $f_{\alpha,j}$ is constant in x_j on I_α and vanishes outside, we only need to consider summands with coarser resolution $\lambda'_j < \lambda$ in that coordinate, and where the support of $\psi_{\boldsymbol{\alpha}}$ contains the support of $f_{\alpha,j}$. That is the case for $\kappa'_j = \lfloor 2^{\lambda'_j - \lambda}\kappa \rfloor$ with $\lambda'_j = -\infty, 0, \dots, \lambda - 1$. For such indices $\boldsymbol{\alpha}' \leftrightarrow (\boldsymbol{\kappa}', \boldsymbol{\lambda}')$ we have

$$\left\langle \psi_{\boldsymbol{\alpha}'},\, f_{\alpha,j}\right\rangle^2 = 2^{\max\{0,\lambda'_j\}-\lambda}\, \langle \psi_{\boldsymbol{\alpha}''},\, f\rangle^2 = 2^{\max\{0,\lambda'_j\}-\lambda}\, \widetilde{f}^2(\boldsymbol{\alpha}'') \,,$$

where $\alpha''_{j'} = \alpha'_{j'}$ for $j' \neq j$, and $\alpha''_j = \alpha$. Hence we obtain

$$\widetilde{f}(\alpha\,\mathbf{e}_j) \ge 2^{\lambda/2} \underbrace{\left(2^{-\lambda} + \sum_{l=0}^{\lambda-1} 2^{l-\lambda}\right)}_{=1} \sum_{\substack{\boldsymbol{\alpha}''\in\mathbb{N}_0^d \\ \alpha''_j = \alpha}} \widetilde{f}^2(\boldsymbol{\alpha}'') \,. \tag{9}$$

Based on this relation between the coefficients, we can estimate

$$\begin{aligned} 1 = \|f\|_{L_2}^2 &= \sum_{\boldsymbol{\alpha}\in\mathbb{N}_0^d} \widetilde{f}^2(\boldsymbol{\alpha}) \geq \sum_{j=1}^{d}\sum_{\lambda=0}^{r-1}\sum_{\kappa=0}^{2^\lambda-1} \widetilde{f}^2((2^\lambda+\kappa)\,\mathbf{e}_j) \\ &\geq \sum_{j=1}^{d}\sum_{\lambda=0}^{r-1} 2^{-\lambda}\left(\sum_{\kappa=0}^{2^\lambda-1} \widetilde{f}((2^\lambda+\kappa)\,\mathbf{e}_j)\right)^2 \\ [(9)] \qquad &\geq \sum_{j=1}^{d}\sum_{\lambda=0}^{r-1}\left(\sum_{\substack{\boldsymbol{\alpha}\in\mathbb{N}_0^d \\ \lambda_j=\lambda}} \widetilde{f}^2(\boldsymbol{\alpha})\right)^2 . \end{aligned}$$

Taking the square root, and using the norm estimate $\|\mathbf{v}\|_1 \leq \sqrt{m}\,\|\mathbf{v}\|_2$ for $\mathbf{v}\in\mathbb{R}^m$, here with $m = dr$, we end up with

$$1 \geq \frac{1}{\sqrt{dr}} \sum_{j=1}^{d}\sum_{\lambda=0}^{r-1}\sum_{\substack{\boldsymbol{\alpha}\in\mathbb{N}_0^d \\ \lambda_j=\lambda}} \widetilde{f}^2(\boldsymbol{\alpha}) = \frac{1}{\sqrt{dr}} \sum_{\substack{\boldsymbol{\alpha}\in\mathbb{N}_0^d \\ \boldsymbol{\lambda}<r}} |\boldsymbol{\alpha}|_0\, \widetilde{f}^2(\boldsymbol{\alpha}) \geq \frac{k+1}{\sqrt{dr}} \sum_{\substack{\boldsymbol{\alpha}\in\mathbb{N}_0^d \\ |\boldsymbol{\alpha}|_0>k \\ \boldsymbol{\lambda}<r}} \widetilde{f}^2(\boldsymbol{\alpha})\,.$$

This proves the lemma. □

By virtue of the above lemma we obtain the following error and complexity bound, compare to Bshouty and Tamon [5, Theorem 5.1] for the Boolean setting. (Their error criterion is slightly different, namely, it is the L_1-error at a prescribed confidence level rather than the expected L_1-distance.)

Theorem 3.3. *For the algorithm $\bar{A}_{n,k,r} = (\bar{A}^{\omega}_{n,k,r})_{\omega\in\Omega}$ as defined in (7) we have the error bound*

$$e(\bar{A}_{n,k,r}, F^d_{\mathrm{mon}}) \;\leq\; 5\,\frac{d}{2^r} + 4\,\frac{\sqrt{dr}}{k+1} + 4\,\frac{\exp[k(1+\log\frac{d}{k}+(\log 2)\,r)]}{n}\,.$$

Given $0<\varepsilon<1$, the ε-complexity for the Monte Carlo approximation of monotone functions is bounded by

$$n^{\mathrm{ran}}(\varepsilon, F^d_{\mathrm{mon}}) \;\leq\; \min\left\{\exp\left[C\,\frac{\sqrt{d}}{\varepsilon}\,\left(1+\log\frac{d}{\varepsilon}\right)^{3/2}\right],\; \exp\left[d\,\log\frac{d}{2\varepsilon}\right]\right\},$$

with some numerical constant $C>0$.

Proof. Thanks to Lemma 3.1, we may restrict to the analysis of the algorithm $\widehat{A}_{n,k,r}$ for sign-valued functions $f\in F^d_{\mathrm{mon}\pm}$.

Since we only take certain coefficients until a resolution r into account, the reconstruction (6) will be a function which is constant on each of 2rd subcubes $C_{r\mathbf{1},\boldsymbol{\kappa}}$ where $\mathbf{1}=(1,\dots,1)$ and $\boldsymbol{\kappa}\in\{0,\dots,2^r-1\}^d$. The algorithm can be seen as actually approximating

$$\operatorname{sgn} f_r\,, \qquad \text{where} \quad f_r := \sum_{\substack{\boldsymbol{\alpha}\in\mathbb{N}_0^d \\ \boldsymbol{\lambda}<r}} \widetilde{f}(\boldsymbol{\alpha})\,\psi_{\boldsymbol{\alpha}}\,.$$

Since on the one hand, the basis functions are constant on each of these 2^{rd} subcubes, and on the other hand, we have 2^{rd} basis functions up to this resolution, the function f_r takes on each of the

subcubes the average function value of f on that subcube, which is between -1 and $+1$. The function sgn f_r takes the subcubewise predominant value of f, which is either -1 or $+1$. That is, for $\mathbf{X}, \mathbf{X}' \sim \mathrm{unif}\, C_{r\mathbf{1},\boldsymbol{\kappa}}$ we have

$$\mathbb{E}|f(\mathbf{X}) - \operatorname{sgn} f_r(\mathbf{X})| = \mathbb{E}|f(\mathbf{X}) - \operatorname{med}' f(\mathbf{X}')| \;\le\; \mathbb{1}[f \text{ not const. on } C_{r\mathbf{1},\boldsymbol{\kappa}}]\,, \tag{10}$$

and for $\mathbf{x} \in C_{r\mathbf{1},\boldsymbol{\kappa}}$ we can estimate

$$|\operatorname{sgn}(f_r(\mathbf{x})) - f_r(\mathbf{x})| \;\le\; \mathbb{1}[f \text{ not const. on } C_{r\mathbf{1},\boldsymbol{\kappa}}]\,. \tag{11}$$

Similarly to the upper bound part within the proof of Theorem 2.1, we group the subcubes along diagonals. By monotonicity, there is at most one subcube within each diagonal where the sign-valued function f jumps from -1 to $+1$, hence (10) and (11) are non-zero but bounded by 1. Now that there are $2^{rd} - (2^r - 1)^d \le d \cdot 2^{r(d-1)}$ diagonals, and the volume of each subcube is 2^{-rd}, we obtain

$$\|f - \operatorname{sgn} f_r\|_{L_1} \;\le\; \frac{d \cdot 2^{r(d-1)}}{2^{rd}} = \frac{d}{2^r}\,, \qquad \text{as well as} \qquad \|\operatorname{sgn} f_r - f_r\|_{L_2}^2 \;\le\; \frac{d}{2^r}\,. \tag{12}$$

We consider the error for estimating the corresponding coefficients. Exploiting independence of the sample points and unbiasedness of the standard Monte Carlo coefficient estimator (4), for $(\boldsymbol{\lambda}, \boldsymbol{\kappa}) \leftrightarrow \boldsymbol{\alpha} \in \mathbb{N}_0^d$ we have

$$\begin{aligned}
\mathbb{E}[\widetilde{h}(\boldsymbol{\alpha}) - \widetilde{f}(\boldsymbol{\alpha})]^2 &= \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} \psi_{\boldsymbol{\alpha}}(\mathbf{X}_i)\, f(\mathbf{X}_i) - \widetilde{f}(\boldsymbol{\alpha})\right]^2 \\
[\mathbf{X}_i \text{ i.i.d.; unbiasedness}] \qquad &= \frac{1}{n}\operatorname{Var}\left[\psi_{\boldsymbol{\alpha}}(\mathbf{X}_1)\, f(\mathbf{X}_1)\right] \;\le\; \frac{1}{n}\|\psi_{\boldsymbol{\alpha}} \cdot f\|_{L_2}^2 \\
[|f| \le 1] \qquad &\le \frac{1}{n}\|\psi_{\boldsymbol{\alpha}}\|_{L_2}^2 = \frac{1}{n}\,.
\end{aligned} \tag{13}$$

This estimate on the quality of the coefficient approximation can be used for estimating L_2-errors. Regarding the approximation $g = \widehat{A}^{\omega}_{n,k,r}(f) = \operatorname{sgn} h$ as defined in (6), from the observation

$$\operatorname{sgn}(f_r(\mathbf{x})) \;\neq\; g(\mathbf{x}) = \operatorname{sgn}(h(\mathbf{x})) \quad \Rightarrow \quad (\operatorname{sgn}(f_r(\mathbf{x})) - h(\mathbf{x}))^2 \ge 1$$

we conclude

$$\|\operatorname{sgn} f_r - g\|_{L_1} \;\le\; 2\|\operatorname{sgn} f_r - h\|_{L_2}^2 \;\le\; 4\,\|\operatorname{sgn} f_r - f_r\|_{L_2}^2 + 4\,\|f_r - h\|_{L_2}^2\,. \tag{14}$$

Then, combining previous estimates, the expected distance between the input f and the approximate reconstruction $g = \widehat{A}^{\omega}_{n,k,r}(f) = \operatorname{sgn} h$ from (6) can be bounded as follows,

$$\begin{aligned}
\mathbb{E}\|f - g\|_{L_1} &\le \|f - \operatorname{sgn} f_r\|_{L_1} + \mathbb{E}\|\operatorname{sgn} f_r - g\|_{L_1} \\
[(14)] \qquad &\le \|f - \operatorname{sgn} f_r\|_{L_1} + 4\,\|\operatorname{sgn} f_r - f_r\|_{L_2}^2 + 4\mathbb{E}\|f_r - h\|_{L_2}^2 \\
[(12), \text{Parseval}] \qquad &\le 5\,\frac{d}{2^r} + 4\left(\sum_{\substack{\boldsymbol{\alpha}\in\mathbb{N}_0^d \\ |\boldsymbol{\alpha}|_0 > k \\ \boldsymbol{\lambda} < r}} \widetilde{f}(\boldsymbol{\alpha})^2 + \sum_{\substack{\boldsymbol{\alpha}\in\mathbb{N}_0^d \\ |\boldsymbol{\alpha}|_0 \le k \\ \boldsymbol{\lambda} < r}} \mathbb{E}[\widetilde{f}(\boldsymbol{\alpha}) - \widetilde{h}(\boldsymbol{\alpha})]^2\right) \\
[\text{Lemma 3.2; (13)}] \qquad &\le 5\,\frac{d}{2^r} + 4\,\frac{\sqrt{d\,r}}{k+1} + 4\,\frac{\#\mathcal{A}}{n}\,,
\end{aligned} \tag{15}$$

where $\mathcal{A}$ is the index set corresponding to the coefficients that are computed,

$$\mathcal{A} := \{\boldsymbol{\alpha} \in \mathbb{N}_0^d \mid |\boldsymbol{\alpha}|_0 \le k \text{ and } \boldsymbol{\lambda} < r\}\,.$$

We can quantify the size of the index set $\mathcal{A}$ for $k \in \{1, \dots, d\}$ by standard estimates,

$$\#\mathcal{A} = \sum_{l=0}^{k} \binom{d}{l} (2^r - 1)^l \;\le\; 2^{rk} \sum_{l=0}^{k} \binom{d}{l} \;\le\; 2^{rk} \left(\frac{\mathrm{e}\, d}{k}\right)^k .$$

This finally yields the error bound for the Monte Carlo method $\widehat{A}_{n,k,r}$ applied to sign-valued functions $f \in F^d_{\mathrm{mon}\pm}$ as stated in the theorem.

Choosing the resolution $r := \lceil \log_2 \frac{15\,d}{\varepsilon} \rceil$ will bound the first term $5\,d \cdot 2^{-r} \le \varepsilon/3$. Selecting $k := \min\left\{\lfloor 12\sqrt{d\,r}/\varepsilon \rfloor,\ d\right\}$ then guarantees $4\sqrt{d\,r}/(k+1) \le \varepsilon/3$, except for the case $k = d$ where we can even ignore the second term from the estimate (15). Finally, the third term $4 \cdot (\#\mathcal{A})/n$ can be bounded from above by $\varepsilon/3$ if we put

$$n := \left\lceil \frac{12}{\varepsilon} \exp\left(k\left(1 + \log\frac{d}{k} + (\log 2)\, r\right)\right)\right\rceil \;\le\; \exp\left[C\,\frac{\sqrt{d}}{\varepsilon}\left(1 + \log\frac{d}{\varepsilon}\right)^{3/2}\right],$$

with some suitable numerical constant $C > 0$. By this choice we obtain the error bound ε we aimed for.

Note that if ε is too small, we can only choose $k = d$ for the algorithm $A_{n,k,r}$. In this case, for the approximation of f_r, we would take 2^{rd} coefficients into account, n would become much bigger in order to achieve the accuracy we aim for. Instead, one can approximate f directly via the deterministic algorithm A^d_m from Theorem 2.1, which is based on $(m-1)^d$ function values on a regular grid. The worst case error is bounded by $e(A^d_m, F^d_{\mathrm{mon}}) \le d/m$. Taking $m := 2^r$, this gives the same bound that we already have for the accuracy at which $\operatorname{sgn} f_r$ approximates f, see (12). So for small ε, which roughly means $\varepsilon \preceq 1/\sqrt{d}$ (modulo logarithmic terms), we take the deterministic upper bound

$$n^{\mathrm{det}}(\varepsilon, F^d_{\mathrm{mon}}) \;\le\; \exp\left(d\,\log\frac{d}{\varepsilon}\right),$$

compare this to Remark 2.2. □

Remark 3.4 (*Violation of Monotonicity*)**.** For the algorithms we analysed, there is no feature which would guarantee that the output function g is a monotonously increasing function. In fact, the analysis of Lemma 3.2 only requires that the function is monotone in each variable, but it is not necessary to know whether it is monotonously increasing or decreasing.

We may think about a scenario where all computed function values are 1, but accidentally they are computed in the lowermost subcube $C_{r\mathbf{1},\mathbf{0}}$ of the domain $[0,1]^d$ at resolution r, and then some function values of the reconstruction g are still negative and violate the assumption of monotonic growth. Namely, for the linear reconstruction h, the value in the uppermost subcube $C_{r\mathbf{1},(2^r-1)\mathbf{1}}$ at resolution r can be written as

$$\sum_{\substack{\boldsymbol{\alpha}\in\{0,1\}^d \\ |\boldsymbol{\alpha}|_0 \le k}} (-1)^{|\boldsymbol{\alpha}|_0} = \sum_{\ell=0}^{k} \binom{d}{\ell} (-1)^\ell .$$

If $k \le d/2$ is odd, this value is negative. Meanwhile, h is positive in $C_{r\mathbf{1},\mathbf{0}}$, hence the monotonicity is violated, $g = \operatorname{sgn} h \notin F^d_{\mathrm{mon}}$.

Remark 3.5 (*Implementation of the Non-Linear Method* $\bar{A}_{n,k,r}$)**.** The algorithm $\bar{A}_{n,k,r}$ as defined in (7) appears rather abstract with the integral within its definition. There is an explicit

way of representing the algorithm, though. Let $\phi^{\omega}_{n,k,r}$ denote the mapping which returns the output $h = \phi^{\omega}_{n,k,r}(\mathbf{y})$ for the linear algorithm $A_{n,k,r}$ from (5) when given the information $N^{\omega}(f) = \mathbf{y} = (y_1, \dots, y_n) = (f(\mathbf{X}_1), \dots, f(\mathbf{X}_n))$. For the reconstruction mapping $\bar{\phi}^{\omega}_{n,k,r}$ used in $\bar{A}_{n,k,r}$ one may proceed as follows:

- Rearrange the information $(\mathbf{X}_1, y_1), \dots, (\mathbf{X}_n, y_n)$ such that $y_1 \le y_2 \le \cdots \le y_n$.
- Define $y_0 := -1$ and $y_{n+1} := +1$, and use the representation

$$g := \bar{\phi}^{\omega}_{n,k,r}(\mathbf{y}) = \frac{1}{2} \sum_{i=0}^{n} (y_{i+1} - y_i)\ \operatorname{sgn} \underbrace{\phi^{\omega}_{n,k,r}(\overbrace{-1, \dots, -1}^{i \text{ times}}, \overbrace{1, \dots, 1}^{(n-i) \text{ times}})}_{=:g_i} .$$

Implementing the nonlinear algorithm $\bar{A}_{n,k,r}$ is more difficult than for the linear algorithm, the cost for processing the collected information may exceed n. There are different models of computation, see for example the book on IBC of Traub et al. [20, p. 30], or Novak and Woźniakowski [16, Sec 4.1.2]. Heinrich and Milla [8, Sec 6.2] point out that for problems with functions as output, the interesting question is not always about a complete picture of the output g, but about effective computation of approximate function values $g(\mathbf{x})$ on demand. It makes sense to distinguish between pre-processing operations and operations on demand.

In our situation, pre-processing is concerned with rearranging the information, for which the expected computational cost is of order $\mathcal{O}(n \log n)$.

The main difficulty when asked to compute a function value $g(\mathbf{x})$ on demand is to compute $g_i(\mathbf{x})$ for $i = 1, \dots, n$. Once we know $g_0(\mathbf{x})$, it will be easier to compute $g_1(\mathbf{x}), g_2(\mathbf{x}), \dots$ in consecutive order because only few coefficients are affected when switching from $y_i = -1$ to $y_i = +1$. Namely, by linearity of $\phi^{\omega}_{n,k,r}$ we have

$$g_i := g_{i-1} - 2\, \phi^{\omega}_{n,k,r}(\mathbf{e}_i)\,,$$

where $\mathbf{e}_i = (\delta_{ij})_{j=1}^{n}$ is the ith unit vector in $\mathbb{R}^n$. Going back to the details of one-dimensional Haar wavelets ψ_{α_j}, observe that

$$\begin{aligned}[\phi^{\omega}_{n,k,r}(\mathbf{e}_i)](\mathbf{x}) &= \frac{1}{n} \sum_{\substack{\boldsymbol{\alpha} \in \mathbb{N}_0^d \\ |\boldsymbol{\alpha}|_0 \le k \\ \boldsymbol{\lambda} < r}} \psi_{\boldsymbol{\alpha}}(\mathbf{X}_i)\, \psi_{\boldsymbol{\alpha}}(\mathbf{x}) = \frac{1}{n} \sum_{\substack{\boldsymbol{\alpha} \in \mathbb{N}_0^d \\ |\boldsymbol{\alpha}|_0 \le k \\ \boldsymbol{\lambda} < r}} \prod_{j=1}^{d} \psi_{\alpha_j}(\mathbf{X}_i(j))\, \psi_{\alpha_j}(x_j) \\ &= \frac{1}{n} \sum_{\substack{\boldsymbol{\beta} \in \{0,1\}^d \\ |\boldsymbol{\beta}|_1 \le k}} \mathbf{Z}^{\boldsymbol{\beta}}\,,\end{aligned}$$

where $Z_j := \sum_{\alpha_j=1}^{2^r-1} \psi_{\alpha_j}(\mathbf{X}_i(j))\, \psi_{\alpha_j}(x_j)$ and $\mathbf{X}_i(j)$ denotes the jth entry of $\mathbf{X}_i \in [0,1]^d$. It is readily checked that

$$Z_j = \begin{cases} 2^r - 1 & \text{if } \lfloor 2^r\, \mathbf{X}_i(j) \rfloor = \lfloor 2^r\, x_j \rfloor, \\ -1 & \text{otherwise,} \end{cases}$$

so a comparison of the first r digits of the binary representation of $\mathbf{X}_i(j)$ and x_j is actually enough for determining Z_j. In the end, we only need the number b of coordinates $j \in \{1, \dots, d\}$ where $\lfloor 2^r\, \mathbf{X}_i(j) \rfloor = \lfloor 2^r\, x_j \rfloor$, and obtain

$$n\, [\phi^{\omega}_{n,k,r}(\mathbf{e}_i)](\mathbf{x}) = \sum_{\ell=0}^{b \wedge k} \binom{b}{\ell} (2^r - 1)^{\ell} \sum_{m=0}^{(d-b) \wedge (k-\ell)} \binom{d-b}{m} (-1)^m \;=:\; \chi(b) \in \mathbb{Z}\,.$$

These values $\chi(b)$ are needed for $b \in \{0, \dots, d\}$. Since they only depend on parameters of the algorithm, they can be prepared before the algorithm is applied to an instance and do not count for the cost of processing the data. From these values one can also compute $g_0(\mathbf{x})$. If we do not want to store the values $[\phi^{\omega}_{n,k,r}(\mathbf{e}_i)](\mathbf{x})$ for $i = 1, \dots, n$, we will need to compute them twice—once in order to evaluate $g_0(\mathbf{x})$, once for calculating the difference between $g_i(\mathbf{x})$ and $g_{i-1}(\mathbf{x})$. Or we only compute them once but need storage for n numbers. In any case, the number of binary and arithmetic operations needed for computing the output $g(\mathbf{x}) := [\bar{\phi}^{\omega}_{n,k,r}(\mathbf{y})](\mathbf{x})$ on demand is of order $\mathcal{O}(drn)$.

4. Intractability of randomized approximation

4.1. The result—Monte Carlo lower bound

As we have seen in Section 3, for the L_1-approximation of monotone functions the dimension dependence can be significantly reduced in the randomized setting, at least if the error threshold $\varepsilon > 0$ is fixed. Within this section, however, we show a lower bound which implies that the problem is still *not weakly tractable* in the randomized setting, we thus call it *intractable*.

For the proof we switch to an average case setting for Boolean functions, an idea that has already been used by Blum, Burch, and Langford [4, Sec 4]. From their result one can already extract that for any fixed $\varepsilon \in (0, 1)$ the Monte Carlo complexity for the approximation of monotone functions depends at least exponentially on $\sqrt{d}$. The focus of Blum et al. was to show that if we admit the information budget n to grow only polynomially in d, the achievable error will approach the initial error at a rate of almost $1/\sqrt{d}$. In contrast, the aim of this paper is to obtain lower complexity bounds for a range of error thresholds ε which is moderately approaching zero as d is growing. This enables us to prove *intractability*. The different focus leads to the necessity of different tools within the corresponding lower bound proof, see [13, Remark 4.8] for a detailed discussion.

Theorem 4.1. *Consider the randomized approximation of monotone functions. There exist constants $\sigma_0, \nu, \varepsilon_0 > 0$ and $d_0 \in \mathbb{N}$ such that for $d \geq d_0$ we have*

$$n^{\mathrm{ran}}(\varepsilon_0, F^d_{\mathrm{mon}}) \;>\; \nu \, \exp(\sigma_0 \sqrt{d}) \,,$$

and moreover, for $\varepsilon_0 \sqrt{d_0/d} \leq \varepsilon \leq \varepsilon_0$ we have

$$n^{\mathrm{ran}}(\varepsilon, F^d_{\mathrm{mon}}) \;>\; \nu \, \exp\left(c \, \frac{\sqrt{d}}{\varepsilon} \right) ,$$

with $c = \sigma_0 \, \varepsilon_0$.

Specifically, for $d \geq d_0 = 100$ and $\varepsilon_0 = \frac{1}{15}$ we have

$$n^{\mathrm{ran}}(\tfrac{1}{15}, F^d_{\mathrm{mon}}) \;>\; 108 \cdot \exp(\sqrt{d} - \sqrt{100}) \,,$$

for $d = 100$ this means $n^{\mathrm{ran}} > 108$. For $\frac{1}{15}\sqrt{100/d} \leq \varepsilon \leq \frac{1}{15}$ we have

$$n^{\mathrm{ran}}(\varepsilon, F^d_{\mathrm{mon}}) \;>\; 108 \cdot \exp\left(\frac{\sqrt{d}}{15\,\varepsilon} - \sqrt{100} \right) .$$

All these lower bounds hold for varying cardinality as well, see [13, Section 4.3]. Before we give the proof in Section 4.2, we discuss a theoretical consequence of the theorem.

Remark 4.2 (*Intractability*)**.** The above theorem shows that the approximation of monotone functions is *not weakly tractable*. Indeed, consider the sequence $(\varepsilon_d)_{d=d_0}^{\infty}$ of error thresholds $\varepsilon_d := \varepsilon_0 \sqrt{d_0/d}$. Then, regarding $n^{\mathrm{ran}}(\varepsilon, F_{\mathrm{mon}}^d)$ as a function $n(\varepsilon, d)$, we observe

$$\limsup_{\varepsilon^{-1}+d\to\infty} \frac{\log n(\varepsilon, d)}{\varepsilon^{-1}+d} \geq \limsup_{d\to\infty} \frac{\log n(\varepsilon_d, d)}{\varepsilon_d^{-1}+d} \geq \lim_{d\to\infty} \frac{\sigma_0\, d/\sqrt{d_0} + \log \nu}{\varepsilon_0^{-1}\sqrt{d/d_0}+d} = \frac{\sigma_0}{\sqrt{d_0}} > 0\,.$$

This contradicts the definition of weak tractability, as defined in the book of Novak and Woźniakowski [16]. Namely, the problem would be called *weakly tractable* iff the first limit superior was zero. We can also put it like this: in our situation $n(\varepsilon_d, d)$ grows exponentially in d despite the fact that ε_d is only moderately decreasing.

Actually, this behaviour has already been known since the paper of Bshouty and Tamon 1996 [5, Thm 5.3.1] on Boolean monotone functions, however, *weak tractability* is a relatively new notion. Their lower bound can be summarized as follows: For moderately decaying error thresholds $\varepsilon_d \preceq (\sqrt{d}\,(1+\log d))^{-1}$ and sufficiently large d, we have

$$n^{\mathrm{ran}}(\varepsilon_d, F_{\mathrm{mon}}^d) \geq c\, 2^d/\sqrt{d}\,,$$

with some numerical constant $c > 0$. Interestingly, the proof is based on purely combinatorial arguments, without applying minimax arguments. From their approach, however, we can only derive a statement for smaller and smaller ε as $d \to \infty$. So the new lower bounds indeed give a more complete picture on the joint (ε, d)-dependence of the complexity. Since the proof of our theorem is based on Boolean functions, actually we have lower bounds for the easier problem of approximating Boolean monotone functions.

4.2. The proof of the Monte Carlo lower bound

This section contains the proof of Theorem 4.1. Key ideas have already been used by Blum et al. [4], albeit only in the context of Boolean functions. Some modifications within the present proof are mere simplifications with the side effect of improved constants, but several changes are substantial and marked as such.

We consider the subclass $F_2^d \subset F_{\mathrm{mon}\pm}^d$ of sign-valued monotone functions which are constant on each of the 2^d subcubes $C_{\mathbf{i}}$, $\mathbf{i} \in \{0,1\}^d$, if we split the domain $[0,1]^d$ just as in the proof of Theorem 2.1 with $m = 2$. Any such function $f \in F_2^d$ is uniquely determined by its function values $f(\mathbf{x})$ in the corners $\mathbf{x} \in \{0,1\}^d$, we have $f|_{C_{\mathbf{x}}} = f(\mathbf{x})$, so effectively we deal with Boolean functions. The lower bound proof for general Monte Carlo methods relies on *Bakhvalov's trick* [2], compare the lower bound part within the proof of Theorem 2.1 for a more basic example of this proof technique. Here now, we construct a probability measure μ on F_2^d and use that for any Monte Carlo algorithm (A_n^ω) we have

$$\begin{aligned} e((A_n^\omega), F_{\mathrm{mon}}^d) &= \sup_{f\in F_{\mathrm{mon}}^d} \mathbb{E}\|A_n^\omega - f\|_{L_1} \geq \int_{F_2^d} \mathbb{E}\|A_n^\omega(f) - f\|_{L_1}\,\mathrm{d}\mu(f) \\ \text{[Fubini]} \qquad &= \mathbb{E}\int_{F_2^d} \|A_n^\omega(f) - f\|_{L_1}\,\mathrm{d}\mu(f) \geq \inf_{A_n} \underbrace{\int_{F_2^d} \|A_n(f) - f\|_{L_1}\,\mathrm{d}\mu(f)}_{=:e(A_n,\mu)}\,, \end{aligned} \tag{16}$$

where the infimum runs over all deterministic algorithms $A_n = \phi \circ N$ that use at most n function values. In order to construct optimal algorithms A_n with regard to minimizing the so-called μ-*average error* $e(A_n, \mu)$, one will need to optimize the output function $g = \phi(\mathbf{y})$ with respect

to the conditional measure $\mu_{\mathbf{y}}$ after knowing information $\mathbf{y} := N(f)$. In our specific situation, which is the L_1-approximation of sign-valued functions, the optimal output is sign-valued as well, taking the pointwise conditional median. The conditional error for this optimal output is given by

$$\begin{aligned}\inf_{g\in L_1}\int_{N^{-1}(\mathbf{y})}\|f-g\|_{L_1}\,\mathrm{d}\mu_{\mathbf{y}}(f) &= 2\int_{[0,1]^d}\min\left\{\mu_{\mathbf{y}}\{f(\mathbf{x})=-1\},\mu_{\mathbf{y}}\{f(\mathbf{x})=+1\}\right\}\,\mathrm{d}\mathbf{x}\\ &= 2^{1-d}\sum_{\mathbf{x}\in\{0,1\}^d}\min\left\{\mu_{\mathbf{y}}\{f(\mathbf{x})=-1\},\mu_{\mathbf{y}}\{f(\mathbf{x})=+1\}\right\}. \end{aligned}\tag{17}$$

We will further use the concept of *augmented information* $\widetilde{y}=\widetilde{N}(f)$ which contains additional knowledge on the input compared to the original information $\mathbf{y}=N(f)$. This will lead to more powerful algorithms with smaller errors, but it is done for the sake of an easier description of the corresponding conditional measure $\mu_{\widetilde{y}}$. The lower bounds we obtain for algorithms with the augmented oracle, a fortiori, are lower bounds for algorithms with the standard oracle.

The proof is organized in seven steps.

Step 1: The general structure of the measure μ on F_2^d.

Step 2: Introduce the augmented information.

Step 3: Estimate the number of points $\mathbf{x}\in\{0,1\}^d$ for which $f(\mathbf{x})$ is still – to some extend – undetermined, even after knowing the augmented information.

Step 4: Further specify the measure μ, and give estimates on the conditional probability for the event $f(\mathbf{x})=-1$ for the set of still fairly uncertain $\mathbf{x}$ from the step before.

Step 5: A general formula for the lower bound.

Step 6: Relate estimates for ε_0 and d_0 to estimates for smaller ε and larger d.

Step 7: Explicit numerical values.

Step 1: General structure of the measure μ.

We define a measure μ on F_2^d that can be represented by a randomly drawn set $U\subseteq W:=\{\mathbf{x}\in\{0,1\}^d\mid |\mathbf{x}|_1=t\}$, with $t\in\mathbb{N}$ being a suitable parameter, and a boundary value $b\in\mathbb{N}$, $t\le b\le d$, namely

$$f_U(\mathbf{x}):=\begin{cases}-1, & \text{if } |\mathbf{x}|_1\le b \text{ and } \nexists\,\mathbf{u}\in U \text{ with } \mathbf{u}\le\mathbf{x},\\ +1, & \text{if } |\mathbf{x}|_1>b \text{ or } \exists\,\mathbf{u}\in U \text{ with } \mathbf{u}\le\mathbf{x}.\end{cases}\tag{18}$$

The boundary value $b\in\mathbb{N}$ will facilitate considerations in connection with the augmented information in Step 2. We draw U such that the $f(\mathbf{w})$ with $\mathbf{w}\in W$ are independent Bernoulli random variables with $p=\mu\{f(\mathbf{w})=+1\}=1-\mu\{f(\mathbf{w})=-1\}$. The parameter $p\in(0,1)$ will be specified in Step 4.

Step 2: Augmented information.

Now, for any (possibly adaptively obtained) info $\mathbf{y}=N(f)=(f(\mathbf{x}_1),\dots,f(\mathbf{x}_n))$ with $\mathbf{x}_i\in\{0,1\}^d$, we define the augmented information

$$\widetilde{y}:=(V_\ominus,V_\oplus),$$

where $V_\ominus \subseteq W \setminus U$ and $V_\oplus \subseteq U$ represent knowledge about the instance f that μ-almost surely implies the information $\mathbf{y}$. We know $f(\mathbf{u}) = -1$ for $\mathbf{u} \in V_\ominus$, and $f(\mathbf{u}) = +1$ for $\mathbf{u} \in V_\oplus$. In detail, let $\le_{\mathrm{L}}$ be the lexicographic order of the elements of W, then $\min_{\mathrm{L}} V$ denotes the first element of a set $V \subseteq W$ with respect to this order. For a single function evaluation $f(\mathbf{x})$ the augmented oracle reveals the sets

$$V_\ominus^{\mathbf{x}} := \begin{cases} \emptyset & \text{if } |\mathbf{x}|_1 > b\,, \\ \{\mathbf{v} \in W \mid \mathbf{v} \le \mathbf{x}\} & \text{if } f(\mathbf{x}) = -1\,, \\ \{\mathbf{v} \in W \mid \mathbf{v} \le \mathbf{x} \text{ and } \mathbf{v} <_{\mathrm{L}} \min_{\mathrm{L}}\{\mathbf{u} \in U \mid \mathbf{u} \le \mathbf{x}\}\} & \text{if } f(\mathbf{x}) = +1 \\ & \text{and } |\mathbf{x}|_1 \le b\,, \end{cases}$$

$$V_\oplus^{\mathbf{x}} := \begin{cases} \emptyset & \text{if } |\mathbf{x}|_1 > b \text{ or } f(\mathbf{x}) = -1\,, \\ \{\min_{\mathrm{L}}\{\mathbf{u} \in U \mid \mathbf{u} \le \mathbf{x}\}\} & \text{if } f(\mathbf{x}) = +1 \text{ and } |\mathbf{x}|_1 \le b\,, \end{cases}$$

and altogether the augmented information is

$$\widetilde{y} = (V_\ominus, V_\oplus) := \left(\bigcup_{i=1}^n V_\ominus^{\mathbf{x}_i}\,,\ \bigcup_{i=1}^n V_\oplus^{\mathbf{x}_i} \right).$$

Note that computing $f(\mathbf{x})$ for $|\mathbf{x}|_1 > b$ is a waste of information, so no algorithm designer would decide to compute such samples. Since $\#V_\ominus^{\mathbf{x}} \le \binom{|\mathbf{x}|_1}{t} \le \binom{b}{t}$ for $|\mathbf{x}|_1 \le b$, and $\#V_\oplus^{\mathbf{x}} \le 1$, we have the estimates

$$\#V_\ominus \le n \binom{b}{t}, \quad \text{and} \quad \#V_\oplus \le n\,. \tag{19}$$

(Blum et al. [4] did not have a boundary value b but used a Chernoff bound for giving a probabilistic bound on $\#V_\ominus$ instead.)

Step 3: Number of points $\mathbf{x} \in \{0,1\}^d$ where $f(\mathbf{x})$ is still fairly uncertain.

For any point $\mathbf{x} \in \{0,1\}^d$ we define the set

$$W_{\mathbf{x}} := \{\mathbf{w} \in W \mid \mathbf{w} \le \mathbf{x}\}$$

of points that are "relevant" to $f(\mathbf{x})$. Given the augmented information $\widetilde{y} = (V_\ominus, V_\oplus)$, we are interested in points where it is not yet clear whether $f(\mathbf{x}) = +1$ or $f(\mathbf{x}) = -1$. In detail, these are points $\mathbf{x}$ where $W_{\mathbf{x}} \cap V_\oplus = \emptyset$, for which $f(\mathbf{x}) = -1$ is still possible. Furthermore, $W_{\mathbf{x}} \setminus V_\ominus$ shall be big enough, say $\#(W_{\mathbf{x}} \setminus V_\ominus) \ge M$ with $M \in \mathbb{N}$, so that the conditional probability $p_{\mathbf{x}} := \mu_{\widetilde{y}}\{f(\mathbf{x}) = +1\}$ is not too small. For our estimates in (26) it will be necessary to restrict to points $|\mathbf{x}|_1 \ge a \in \mathbb{N}$, we suppose $t \le a \le b$. The set of all these points shall be denoted by

$$B := \{\mathbf{x} \in D_{ab} \mid W_{\mathbf{x}} \cap V_\oplus = \emptyset,\ \#(W_{\mathbf{x}} \setminus V_\ominus) \ge M\}\,,$$
$$\text{where} \quad D_{ab} := \{\mathbf{x} \in \{0,1\}^d \mid a \le |\mathbf{x}|_1 \le b\}\,.$$

We aim to find a lower bound for the cardinality of B. Within the proof of Blum et al. [4] Hoeffding bounds were used. We will employ the Berry–Esseen inequality on the speed of convergence of the Central Limit Theorem, instead, and it is only with Berry–Esseen that we can draw conclusions for small ε, as it is done in Step 6.

Proposition 4.3 (*Berry–Esseen Inequality*)**.** *Let $Z_1, Z_2, \ldots$ be i.i.d. random variables with zero mean, unit variance and finite third absolute moment β_3. Then there exists a universal*

constant C_0 such that

$$\left| \mathbb{P}\left\{ \frac{1}{\sqrt{d}} \sum_{j=1}^{d} Z_j \le x \right\} - \Phi(x) \right| \le \frac{C_0\, \beta_3}{\sqrt{d}}\,,$$

where $\Phi(\cdot)$ is the cumulative distribution function of the univariate standard normal distribution.

The best estimates known on C_0 are

$$C_E := \frac{\sqrt{10}+3}{6\sqrt{2\pi}} = 0.409732\ldots \le C_0 < 0.4748$$

see Shevtsova [19].

Step 3.1: Bounding $\#D_{a,b}$.

Let $a := \lceil \frac{d}{2} + \alpha \frac{\sqrt{d}}{2} \rceil$ and $b := \lfloor \frac{d}{2} + \beta \frac{\sqrt{d}}{2} \rfloor$ with $\beta - \alpha \ge 2/\sqrt{d}$, hence $a \le b$. Consider Rademacher random variables $Z_1, \ldots, Z_d \overset{\text{iid}}{\sim} \text{unif}\{-1, +1\}$. Note that the Z_j have zero mean, unit variance, and third absolute moment $\beta_3 = 1$. Applying Proposition 4.3 twice to the Z_j, we obtain

$$\begin{aligned} \frac{\#D_{ab}}{\#\{0,1\}^d} &= \frac{1}{2^d} \sum_{k=a}^{b} \binom{d}{k} = \mathbb{P}\left\{ \alpha \le \frac{1}{\sqrt{d}} \sum_{j=1}^{d} Z_j \le \beta \right\} \\ &\ge \underbrace{\Phi(\beta) - \Phi(\alpha)}_{=:C_{\alpha\beta}} - \frac{2\,C_0}{\sqrt{d}} =: r_0(\alpha, \beta, d)\,. \end{aligned} \tag{20}$$

Step 3.2: The influence of $\mathbf{w} \in W$ ***(in particular*** $\mathbf{w} \in V_\oplus$***).***

(This step becomes essential for small ε in Step 6. For the focus of Blum et al. [4] with ε close to the initial error, the trivial estimate $\#(Q_{\mathbf{w}} \cap D_{ab})/\#Q_{\mathbf{w}} \le 1$ suffices.) Now, let $t := \lceil \tau \sqrt{d} \rceil$ with $\tau > 0$, and for $\mathbf{w} \in W$ define

$$Q_{\mathbf{w}} := \{\mathbf{x} \in \{0,1\}^d \mid \mathbf{w} \le \mathbf{x}\}\,,$$

this is the set of all points inside the area of influence of $\mathbf{w}$. Similarly to Step 3.1, we obtain

$$\begin{aligned} \frac{\#(Q_{\mathbf{w}} \cap D_{ab})}{\#Q_{\mathbf{w}}} &= \frac{\#\{\mathbf{x} \in \{0,1\}^{d-t} \mid a - t \le |\mathbf{x}|_1 \le b - t\}}{2^{d-t}} \\ &= \frac{1}{2^{d-t}} \sum_{k=a-t}^{b-t} \binom{d-t}{k} \\ \text{[Proposition 4.3]} \quad &\le \Phi\left(\frac{2b-t}{\sqrt{d-t}} \right) - \Phi\left(\frac{2a-t}{\sqrt{d-t}} \right) + \frac{2\,C_0}{\sqrt{d-t}} \\ \text{[(23), (24)]} \quad &\le \left[\underbrace{\Phi(\beta - \tau) - \Phi(\alpha - \tau)}_{=:C_{\alpha\beta\tau}} + \underbrace{\left(\frac{1}{\sqrt{2\pi}} + 2\,C_0 \right)}_{=:C_1} \frac{1}{\sqrt{d}} \right] \frac{1}{\sqrt{1 - t/d}}\,, \end{aligned} \tag{21}$$

where for $\tau < \sqrt{d} - 1/\sqrt{d}$ we have

$$1 \le \frac{1}{\sqrt{1 - t/d}} \le \frac{1}{\sqrt{1 - \tau/\sqrt{d} - 1/d}} =: \kappa_\tau(d) \xrightarrow[d\to\infty]{} 1\,. \tag{22}$$

Within the above calculation (21), we exploited that the density of the Gaussian distribution is decreasing with growing distance to 0, in detail, for $t_0 < t_1$ and $\kappa \geq 1$ we have

$$\begin{aligned}\Phi(\kappa\, t_1) - \Phi(\kappa\, t_0) &= \frac{1}{\sqrt{2\pi}} \int_{\kappa\, t_0}^{\kappa\, t_1} \exp\left(-\frac{t^2}{2}\right) \mathrm{d}t = \frac{\kappa}{\sqrt{2\pi}} \int_{t_0}^{t_1} \exp\left(-\frac{\kappa^2\, s^2}{2}\right) \mathrm{d}s \\ &\leq \frac{\kappa}{\sqrt{2\pi}} \int_{t_0}^{t_1} \exp\left(-\frac{s^2}{2}\right) \mathrm{d}s = \kappa\, [\Phi(t_1) - \Phi(t_0)]\,. \end{aligned} \tag{23}$$

Namely, we took $\kappa = 1/\sqrt{1 - t/d}$ which comes from replacing $1/\sqrt{d - t}$ by $1/\sqrt{d}$. Furthermore, we shifted the Φ-function, knowing that its derivative takes values between 0 and $1/\sqrt{2\pi}$, so for $t_0 < t_1$ and $\delta \in \mathbb{R}$ we have

$$\Big|\big[\Phi(t_1 + \delta) - \Phi(t_0 + \delta)\big] - \big[\Phi(t_1) - \Phi(t_0)\big]\Big| \;\leq\; \frac{|\delta|}{\sqrt{2\pi}}\,, \tag{24}$$

in our case $\delta = t/\sqrt{d} - \tau \leq 1/\sqrt{d}$.

Step 3.3: The influence of $V_\ominus$.
Markov's inequality gives

$$\sum_{\mathbf{w} \in V_\ominus} \#(Q_{\mathbf{w}} \cap D_{ab}) = \sum_{\mathbf{x} \in D_{ab}} \#(W_{\mathbf{x}} \cap V_\ominus) \geq N\, \#\{\mathbf{x} \in D_{ab} \mid \#(W_{\mathbf{x}} \cap V_\ominus) \geq N\}\,, \tag{25}$$

with $N \in \mathbb{N}$. Using this, we can carry out the estimate

$$\begin{aligned} \#\{\mathbf{x} \in D_{ab} \mid \#(W_{\mathbf{x}} \setminus V_\ominus) \geq M\} &= \#\{\mathbf{x} \in D_{ab} \mid \#(W_{\mathbf{x}} \cap V_\ominus) \leq \#W_{\mathbf{x}} - M\} \\ &\geq \#\{\mathbf{x} \in D_{ab} \mid \#(W_{\mathbf{x}} \cap V_\ominus) \leq \binom{a}{t} - M\} \\ &= \#D_{ab} - \#\{\mathbf{x} \in D_{ab} \mid \#(W_{\mathbf{x}} \cap V_\ominus) > \binom{a}{t} - M\} \\ [(25)] \quad &\geq \#D_{ab} - \frac{1}{\binom{a}{t} - M + 1} \sum_{\mathbf{w} \in V_\ominus} \#(Q_{\mathbf{w}} \cap D_{ab})\,. \end{aligned} \tag{26}$$

Step 3.4: Final estimates on $\#B$.
Putting all this together, we estimate the cardinality of B:

$$\begin{aligned} \frac{\#B}{\#\{0,1\}^d} &= \frac{\#\left(\{\mathbf{x} \in D_{ab} \mid \#(W_{\mathbf{x}} \setminus V_\ominus) \geq M\} \setminus \bigcup_{\mathbf{w} \in V_\oplus} Q_{\mathbf{w}}\right)}{\#\{0,1\}^d} \\ [(26), \text{any } \mathbf{w} \in W] \quad &\geq \frac{\#D_{ab}}{\#\{0,1\}^d} \\ &\quad - \frac{\#Q_{\mathbf{w}}}{\#\{0,1\}^d} \left(\frac{\#V_\ominus}{\binom{a}{t} - M + 1} + \#V_\oplus\right) \frac{\#(Q_{\mathbf{w}} \cap D_{ab})}{\#Q_{\mathbf{w}}} \\ [(19), (20), (21)+(22)] \quad &\geq C_{\alpha\beta} - \frac{2\, C_0}{\sqrt{d}} \\ &\quad - n\, 2^{-t} \left(\frac{\binom{b}{t}}{\binom{a}{t} - M + 1} + 1\right) \left[C_{\alpha\beta\tau} + \frac{C_1}{\sqrt{d}}\right] \kappa_\tau(d)\,. \end{aligned}$$

Assuming $\alpha - 2\tau \geq -\sqrt{d} + 2/\sqrt{d}$ will guarantee $t < a$. We estimate the ratio

$$\binom{b}{t}\Big/\binom{a}{t} \leq \left(\frac{a+1}{a-t+1}\right)^{b-a} \leq \left(\frac{\frac{d}{2} + \alpha\frac{\sqrt{d}}{2} + 1}{\frac{d}{2} + (\alpha - 2\tau)\frac{\sqrt{d}}{2}}\right)^{(\beta-\alpha)\sqrt{d}/2}$$
$$\leq \exp\Bigg((\beta-\alpha)\,\tau\,\underbrace{\left(1 + \frac{\alpha - 2\tau}{\sqrt{d}}\right)^{-1}}_{=:\kappa_{\alpha\tau}(d)} + \underbrace{\frac{\beta-\alpha}{\sqrt{d} + \alpha - 2\tau}}_{=:K_{\alpha\beta\tau}(d)}\Bigg) =: \sigma_{\alpha\beta\tau}(d)\,, \tag{27}$$

where we have $1 \leq \kappa_{\alpha\tau}(d) \xrightarrow[d\to\infty]{} 1$ and $0 \leq K_{\alpha\beta\tau}(d) \xrightarrow[d\to\infty]{} 0$. (Note that the above estimate is asymptotically optimal, $1 \leq \binom{b}{t}/\binom{a}{t} \xrightarrow[d\to\infty]{} \exp((\beta-\alpha)\,\tau)$.) We finally choose the information cardinality $n = \lfloor \nu 2^t \rfloor$, and put $M := \lceil \lambda \binom{a}{t} \rceil$ with $0 < \lambda < 1$, so that we obtain the estimate

$$\frac{\#B}{\#\{0,1\}^d} \geq \underbrace{\left[C_{\alpha\beta} - \frac{2\,C_0}{\sqrt{d}}\right]}_{=r_0(\alpha,\beta,d)} - \nu\,\underbrace{\left(\frac{\sigma_{\alpha\beta\tau}(d)}{1-\lambda} + 1\right)\left[C_{\alpha\beta\tau} + \frac{C_1}{\sqrt{d}}\right]\kappa_\tau(d)}_{=:r_1(\alpha,\beta,\tau,\lambda,d)}$$
$$=:\ r_B(\alpha,\beta,\tau,\lambda,\nu,d)\,. \tag{28}$$

With all the other conditions on the parameters imposed before, for sufficiently large d we will have $r_0(\ldots) > 0$. Furthermore, we always have $r_1(\ldots) > 0$, so choosing $0 < \nu < r_0(\ldots)/r_1(\ldots)$ will guarantee $r_B(\ldots)$ to be positive.

Step 4: Specification of μ and bounding conditional probabilities.

We specify the measure μ on the set of functions $\{f_U \mid U \subseteq W\} \subset F_2^d$ defined as in (18). Recall that the $f(\mathbf{w})$ (for $\mathbf{w} \in W$) shall be independent Bernoulli random variables with probability $p = \mu\{f(\mathbf{w}) = +1\}$. Knowing the augmented information $\widetilde{y} = (V_\ominus, V_\oplus)$, the values $f(\mathbf{w})$ are still independent random variables with conditional probabilities

$$\mu_{\widetilde{y}}\{f(\mathbf{w}) = +1\} = \begin{cases} 0 & \text{if } \mathbf{w} \in V_\ominus, \\ 1 & \text{if } \mathbf{w} \in V_\oplus, \\ p & \text{if } \mathbf{w} \in W \setminus (V_\ominus \cup V_\oplus). \end{cases}$$

Then for $\mathbf{x} \in B$ we have the estimate

$$\mu_{\widetilde{y}}\{f(\mathbf{x}) = -1\} \ \leq\ (1-p)^M \ \leq\ \exp\left(-p\lambda\binom{a}{t}\right) = \exp(-\lambda\varrho)\,,$$

where we write $p := \varrho/\binom{a}{t}$ with $0 < \varrho < \binom{a}{t}$. The other estimate is

$$\mu_{\widetilde{y}}\{f(\mathbf{x}) = -1\} \geq (1-p)^{\binom{b}{t}} = \exp\left(\log(1-p)\binom{b}{t}\right)$$
$$\geq \exp\Bigg(-\varrho\,\sigma_{\alpha\beta\tau}(d)\left(\underbrace{\frac{1}{2} + \frac{1}{2\,(1-\varrho/\gamma_{\alpha\tau}(d))}}_{=:\kappa_{\varrho\gamma}(d)}\right)\Bigg) =: q_0(\alpha,\beta,\tau,\varrho,d)$$
$$\xrightarrow[d\to\infty]{} \exp\Big(-\varrho\,\exp((\beta-\alpha)\,\tau)\Big)\,. \tag{29}$$

Here we used that, for $0 \leq p < 1$,

$$0 \geq \log(1-p) = -\left(p + \sum_{k=2}^{\infty}\frac{p^k}{k}\right) \geq -\left(p + \sum_{k=2}^{\infty}\frac{p^k}{2}\right) = -p\left(\frac{1}{2} + \frac{1}{2\,(1-p)}\right),$$

together with the estimates

$$p\binom{b}{t} \le \varrho\,\sigma_{\alpha\beta\tau}(d)\,,$$

and

$$\binom{a}{t} \ge \left(\frac{a}{t}\right)^t \ge \left(\frac{\sqrt{d}+\alpha}{2\left(\tau+1/\sqrt{d}\right)}\right)^{\tau\sqrt{d}} =: \gamma_{\alpha\tau}(d) \ge 1\,. \tag{30}$$

The last estimate (30) relies on the constraint $\alpha - 2\tau \ge -\sqrt{d} + 2/\sqrt{d}$ (and hence $t < a$). Note that $\gamma_{\alpha\tau}(d) \xrightarrow[d\to\infty]{} \infty$ implies $\kappa_{\varrho\gamma}(d) \xrightarrow[d\to\infty]{} 1$. It follows that for $\mathbf{x} \in B$,

$$\begin{aligned}&\min\Big\{\mu_{\widetilde{y}}\{f(\mathbf{x}) = +1\},\ \mu_{\widetilde{y}}\{f(\mathbf{x}) = -1\}\Big\}\\ &\quad \ge \min\left\{1 - \exp\left(-\varrho\,\lambda\right),\ q_0(\alpha,\beta,\tau,\varrho,d)\right\} =: q(\alpha,\beta,\tau,\lambda,\varrho,d)\,.\end{aligned} \tag{31}$$

Step 5: The final error bound.

By Bakhvalov's trick (16) we obtain the final estimate for $n \le \nu\, 2^{\tau\sqrt{d}} = \nu\, \exp(\sigma\sqrt{d})$, where $\sigma = \tau\, \log 2$,

$$\begin{aligned} e^{\mathrm{ran}}(n, F^d_{\mathrm{mon}}) &\ge \inf_{A_n} e(A_n, \mu)\\ \text{[(17) for any valid } \widetilde{y}] \quad &\ge 2\,\frac{\#B}{\#\{0,1\}^d}\,\min\Big\{\mu_{\widetilde{y}}\{f(\mathbf{x}) = 0\},\ \mu_{\widetilde{y}}\{f(\mathbf{x}) = 1\} \mid \mathbf{x} \in B\Big\}\\ \text{[(28) and (31)]} \quad &\ge 2\, r_B(\alpha,\beta,\tau,\lambda,\nu,d)\cdot q(\alpha,\beta,\tau,\lambda,\varrho,d)\\ &=: \widehat{\varepsilon}(\alpha,\beta,\tau,\lambda,\nu,\varrho,d)\,. \end{aligned} \tag{32}$$

Fixing $d = d_0$, and with appropriate values for the other parameters as discussed in Step 3.4, we can provide $r_B(\ldots) > 0$. The value of ϱ should be adapted for that $q(\ldots)$ is big (and positive in the first place). The function $\widehat{\varepsilon}(\ldots, d)$ is monotonically increasing in d, so an error bound for $d = d_0$ implies error bounds for $d \ge d_0$ while keeping in particular ν and τ. Clearly, for any $0 < \varepsilon_0 < \widehat{\varepsilon}(\ldots)$, this gives lower bounds for the information complexity,

$$n^{\mathrm{ran}}(\varepsilon_0, F^d_{\mathrm{mon}}) \;>\; \nu\, \exp(\sigma\sqrt{d})\,, \qquad \text{for } d \ge d_0.$$

Step 6: Smaller ε and bigger exponent τ for higher dimensions.

The following sophisticated considerations lead to refined results compared to those of Blum et al. [4]. If we have a lower bound $\widehat{\varepsilon}(\alpha_0,\beta_0,\tau_0,\lambda,\nu,\varrho,d_0) > \varepsilon_0$, then for $d \ge d_0$ and $\tau_0 \le \tau \le \tau_0\sqrt{d/d_0}$ we obtain the lower bound

$$\widehat{\varepsilon}(\alpha(\tau),\beta(\tau),\tau,\lambda,\nu,\varrho,d) \;>\; \varepsilon_0\,\frac{\tau_0}{\tau} =: \varepsilon \tag{33}$$

with $\alpha(\tau) = \alpha_0\,\frac{\tau_0}{\tau}$ and $\beta(\tau) = \beta_0\,\frac{\tau_0}{\tau}$, supposing the additional conditions $\beta_0 \le \tau_0$ and $-\tau_0 \le \alpha_0 \le 0$. This provides the estimate

$$n^{\mathrm{ran}}(\varepsilon, F^d_{\mathrm{mon}}) \ge \nu\, 2^{\tau\sqrt{d}} = \nu\, 2^{\tau_0\,\varepsilon_0\,\sqrt{d}/\varepsilon}\,,$$

valid under the constraint $\varepsilon_0\,\sqrt{d_0/d} \le \varepsilon \le \varepsilon_0$. This is reflected in the statement of Theorem 4.1 with $c = \tau_0\varepsilon_0\log 2$.

In detail, showing (33) can be split into proving inequalities for the factors of $\widehat{\varepsilon}(\dots)$ as defined in (32), namely

$$q(\alpha(\tau),\beta(\tau),\tau,\lambda,\varrho,d) \geq q(\alpha_0,\beta_0,\tau_0,\lambda,\varrho,d_0)\,, \qquad \text{and} \tag{34}$$

$$r_B(\alpha(\tau),\beta(\tau),\tau,\lambda,\nu,d) \geq \frac{\tau_0}{\tau}\, r_B(\alpha_0,\beta_0,\tau_0,\lambda,\nu,d_0)\,. \tag{35}$$

Both factors contain the term $\sigma_{\alpha\beta\tau}(d)$ defined in (27). With the given choice of $\alpha(\tau)$ and $\beta(\tau)$, the product $(\beta-\alpha)\tau = (\beta_0-\alpha_0)\tau_0$ is kept constant, which is the key element for the estimate

$$\sigma_{\alpha\beta\tau}(d) \;\leq\; \sigma_{\alpha_0,\beta_0,\tau_0}(d_0)\,. \tag{36}$$

Here we also need

$$1 \;\leq\; \kappa_{\alpha\tau}(d) = \left(1+\frac{\alpha_0\frac{\tau_0}{\tau}-2\tau}{\sqrt{d}}\right)^{-1} \;\leq\; \left(1+\frac{\alpha_0-2\tau_0}{\sqrt{d_0}}\right)^{-1} = \kappa_{\alpha_0,\tau_0}(d_0)\,,$$

as well as

$$\begin{aligned} 0 \;\leq\; K_{\alpha\beta\tau}(d) &= \frac{(\beta_0-\alpha_0)\frac{\tau_0}{\tau}}{\sqrt{d}+\alpha_0\frac{\tau_0}{\tau}-2\tau} \;\leq\; \frac{\beta_0-\alpha_0}{\sqrt{d}+(\alpha_0-2\tau_0)\sqrt{d/d_0}} \\ &\leq\; \frac{\beta_0-\alpha_0}{\sqrt{d_0}+\alpha_0-2\tau_0} = K_{\alpha_0,\beta_0,\tau_0}(d_0)\,, \end{aligned}$$

where we used $\tau_0/\tau \leq 1 \leq \sqrt{d/d_0}$ combined with $\alpha_0 \leq 0$, and $\tau \leq \tau_0\sqrt{d/d_0}$.

Showing (34), by definition of $q(\dots)$ in (31), means examining $q_0(\alpha(\tau),\beta(\tau),\tau,\varrho,d)$, see (29). Knowing (36), the remaining consideration is

$$\begin{aligned} \gamma_{\alpha\tau}(d) &= \left(\frac{\sqrt{d}+\alpha_0\frac{\tau_0}{\tau}}{2\left(\tau+1/\sqrt{d}\right)}\right)^{\tau\sqrt{d}} \geq \left(\frac{\sqrt{d}+\alpha_0\sqrt{d/d_0}}{2\left(\tau_0+\sqrt{d_0}/d\right)\sqrt{d/d_0}}\right)^{\tau_0\sqrt{d_0}} \\ &\geq \left(\frac{\sqrt{d_0}+\alpha_0}{2\left(\tau_0+1/\sqrt{d_0}\right)}\right)^{\tau_0\sqrt{d_0}} = \gamma_{\alpha_0,\tau_0}(d_0) \geq 1\,, \end{aligned}$$

once more using $\tau_0/\tau \leq 1 \leq \sqrt{d/d_0}$ combined with $\alpha_0 \leq 0$, and $\tau \leq \tau_0\sqrt{d/d_0}$.

Showing (35) is more complicated, in view of the definition of $r_B(\dots)$ in (28), we need estimates on $C_{\alpha\beta}$, $C_{\alpha\beta\tau}$ and $\kappa_\tau(d)$. The easiest part is the correcting factor $\kappa_\tau(d)$, see (22), for which by virtue of $\tau \leq \tau_0\sqrt{d/d_0}$ and $d \geq d_0$ we have

$$1 \;\leq\; \kappa_\tau(d) = \left(1-\tau/\sqrt{d}-1/d\right)^{-1/2} \;\leq\; \left(1-\tau_0/\sqrt{d_0}-1/d_0\right)^{-1/2} = \kappa_{\tau_0}(d_0)\,.$$

For the other terms we need to take a detailed look at Gaussian integrals. First we have

$$\begin{aligned} C_{\alpha\beta} &= \frac{1}{\sqrt{2\pi}}\int_\alpha^\beta \exp\left(-\frac{x^2}{2}\right)\mathrm{d}x \\ [x=\tfrac{\tau_0}{\tau}u] \quad &= \frac{\tau_0}{\tau\sqrt{2\pi}}\int_{\alpha_0}^{\beta_0} \exp\left(-\left(\frac{\tau_0}{\tau}\right)^2\frac{u^2}{2}\right)\mathrm{d}u \\ [\tau\geq\tau_0] \quad &\geq \frac{\tau_0}{\tau\sqrt{2\pi}}\int_{\alpha_0}^{\beta_0} \exp\left(-\frac{u^2}{2}\right)\mathrm{d}u = \frac{\tau_0}{\tau}\,C_{\alpha_0,\beta_0}\,. \end{aligned}$$

The second Gaussian integral is a bit trickier,

$$C_{\alpha\beta\tau} = \frac{1}{\sqrt{2\pi}}\int_{\alpha-\tau}^{\beta-\tau} \exp\left(-\frac{x^2}{2}\right)\mathrm{d}x$$

$$\begin{aligned}
[\text{subst. } x+\tau = \frac{\tau_0}{\tau}(u+\tau_0)] \quad &= \frac{\tau_0}{\tau\sqrt{2\pi}} \int_{\alpha_0-\tau_0}^{\beta_0-\tau_0} \exp\left(-\frac{1}{2}\left(\frac{\tau_0}{\tau}(u+\tau_0)-\tau\right)^2\right) \mathrm{d}u \\
[\tfrac{\tau_0}{\tau}(u+\tau_0)-\tau \le u \le 0] \quad &\le \frac{\tau_0}{\tau\sqrt{2\pi}} \int_{\alpha_0-\tau_0}^{\beta_0-\tau_0} \exp\left(-\frac{u^2}{2}\right) \mathrm{d}u = \frac{\tau_0}{\tau} C_{\alpha_0,\beta_0,\tau_0}\,.
\end{aligned}$$

Here, $u \le 0$ followed from the upper integral boundary $u \le \beta_0 - \tau_0$ and the assumption $\beta_0 \le \tau_0$. The other constraint, $\psi(\tau) := \frac{\tau_0}{\tau}(u+\tau_0) - \tau \le u$, followed from $\psi(\tau_0) = u$ and the monotonous decay of $\psi(\tau)$ for $\tau \ge \tau_0$:

$$\begin{aligned}
\psi'(\tau) &= -\frac{\tau_0}{\tau^2}(u+\tau_0) - 1 \\
[\alpha_0 - \tau_0 \le u] \qquad &\le -\frac{\tau_0}{\tau^2}\alpha_0 - 1 \\
[\alpha_0 \ge -\tau_0] \qquad &\le \frac{\tau_0^2}{\tau^2} - 1 \\
[\tau \ge \tau_0] \qquad &\le 0\,.
\end{aligned}$$

Indeed, these estimates on $\kappa_\tau(d)$, $C_{\alpha\beta}$, and $C_{\alpha\beta\tau}$, together with the condition $d \ge d_0$, prove (35).

Step 7: Example for numerical values.

The stated numerical values result from the setting $\alpha_0 = -0.33794$, $\beta_0 = 0.46332$, $\tau_0 = 1.47566 > \frac{1}{\log 2}$ and $\lambda = 0.77399$. We adapt $\varrho = 0.25960$, and for starting dimension $d_0 = 100$ and information budget $n_0 = 108$ (choosing $\nu = n_0 \cdot 2^{-\tau_0\sqrt{d_0}}$ accordingly) we obtain the lower error bound $\widehat{\varepsilon}(\ldots) = 0.0666667\ldots > \frac{1}{15} =: \varepsilon_0$. □

One might try to find different values for different d_0 and n_0, but since reasonable lower bounds start with $d_0 = 100$ while the implementation of algorithms seems hopeless in that dimension, the result should be seen as rather theoretic.

Acknowledgments

This paper is based on a chapter of the author's PhD thesis [13], which has been financially supported by the DFG, Germany Research Training Group 1523. I wish to express my deepest gratitude to my PhD supervisor Erich Novak for his advice during my work on these results. I also wish to acknowledge valuable hints to literature by Mario Ullrich. Last but not least, I wish to thank the referees for carefully reading the manuscript and many detailed hints.

References

[1] G. Alberti, L. Ambrosio, A geometrical approach to monotone functions in $\mathbb{R}^n$, Math. Z. 230 (1999) 259–316.
[2] N.S. Bakhvalov, On the approximate calculation of multiple integrals, Vestnik MGU, Ser. Math. Mech. Astron. Phys. Chem 4 (1959) 3–18 (in Russian). English translation: J. Complexity 31 (4) (2015) 502–516.
[3] R. Bellman, Dynamic Programming, Princeton University Press, 1957.
[4] A. Blum, C. Burch, J. Langford, On learning monotone Boolean functions, in: Proceedings of the 39th Annual Symposium on Foundations of Computer Science, 1998, pp. 408–415.
[5] N.H. Bshouty, C. Tamon, On the Fourier spectrum of monotone functions, J. ACM 43 (4) (1996) 747–770.
[6] K. Chandrasekaran, A. Deshpande, S. Vempala, Sampling s-concave functions: the limit of convexity based isoperimetry, in: Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, in: Lecture Notes in Computer Science, vol. 5687, 2009, pp. 420–433.
[7] F. Gao, J.A. Wellner, Entropy estimate for high-dimensional monotonic functions, J. Multivariate Anal. 98 (2007) 1751–1764.
[8] S. Heinrich, B. Milla, The randomized complexity of indefinite integration, J. Complexity 27 (3–4) (2011) 352–382.

[9] A. Hinrichs, E. Novak, M. Ullrich, H. Woźniakowski, The curse of dimensionality for numerical integration of smooth functions, Math. Comp. 83 (290) (2014) 2853–2863.
[10] A. Hinrichs, E. Novak, H. Woźniakowski, The curse of dimensionality for the class of monotone functions and for the class of convex functions, J. Approx. Theory 163 (8) (2011) 955–965.
[11] C. Katscher, E. Novak, K. Petras, Quadrature formulas for multivariate convex functions, J. Complexity 12 (1) (1996) 5–16.
[12] K.A. Kopotun, Approximation of k-monotone functions, J. Approx. Theory 94 (1998) 481–493.
[13] R.J. Kunsch, High-Dimensional Function Approximation: Breaking the Curse with Monte Carlo Methods (Ph.D. thesis), FSU Jena, 2017, available on arXiv:1704.08213 [math.NA].
[14] E. Novak, Quadrature formulas for monotone functions, Proc. Amer. Math. Soc. 115 (1) (1992) 59–68.
[15] E. Novak, K. Petras, Optimal stochastic quadrature formulas for convex functions, BIT Numer. Math. 34 (2) (1994) 288–294.
[16] E. Novak, H. Woźniakowski, Tractability of Multivariate Problems, Volume I, Linear Information, European Mathematical Society, 2008.
[17] A. Papageorgiou, Integration of monotone functions of several variables, J. Complexity 9 (2) (1993) 252–268.
[18] H. Robbins, S. Monro, A stochastic approximation method, Ann. Math. Stat. 22 (3) (1951) 400–407.
[19] I. Shevtsova, On the absolute constants in the Berry–Esseen type inequalities for identically distributed summands, 2011, Available on arXiv:1111.6554 [math.PR].
[20] J.F. Traub, G.W. Wasilkowski, H. Woźniakowski, Information-Based Complexity, Academic Press, 1988.
[21] J. Vybíral, Weak and quasi-polynomial tractability of approximation of infinitely differentiable functions, J. Complexity 30 (2014) 48–55.