frontiers
in Applied Mathematics and Statistics

ORIGINAL RESEARCH
published: 07 September 2021
doi: 10.3389/fams.2021.702486

A Block-Sparse Tensor Train Format for Sample-Efficient High-Dimensional Polynomial Regression

Michael Götte, Reinhold Schneider and Philipp Trunschke *

Department of Mathematics, Technische Universität Berlin, Berlin, Germany

Low-rank tensors are an established framework for the parametrization of multivariate polynomials. We propose to extend this framework by including the concept of block-sparsity to efficiently parametrize homogeneous, multivariate polynomials with low-rank tensors. This provides a representation of general multivariate polynomials as a sum of homogeneous, multivariate polynomials, represented by block-sparse, low-rank tensors. We show that this sum can be concisely represented by a single block-sparse, low-rank tensor.

We further prove cases, where low-rank tensors are particularly well suited by showing that for banded symmetric tensors of homogeneous polynomials the block sizes in the block-sparse multivariate polynomial space can be bounded independent of the number of variables.

We showcase this format by applying it to high-dimensional least squares regression problems where it demonstrates improved computational resource utilization and sample efficiency.

Keywords: sample efficiency, homogeneous polynomials, sparse tensor networks, alternating least square, empirical L2 approximation

OPEN ACCESS

Edited by:
Edoardo Angelo Di Napoli,
Helmholtz-Verband Deutscher Forschungszentren (HZ), Germany

Reviewed by:
Mazen Ali,
École centrale de Nantes, France
Katharina Kormann,
Uppsala University, Sweden
Antonio Falco,
Universidad CEU Cardenal Herrera, Spain

****Correspondence:***
Philipp Trunschke
ptrunschke@mail.tu-berlin.de

Specialty section:
This article was submitted to Mathematics of Computation and Data Science, a section of the journal Frontiers in Applied Mathematics and Statistics

Received: *29 April 2021*
Accepted: *21 July 2021*
Published: *07 September 2021*

Citation:
Götte M, Schneider R and Trunschke P (2021) A Block-Sparse Tensor Train Format for Sample-Efficient High-Dimensional Polynomial Regression. Front. Appl. Math. Stat. 7:702486. doi: 10.3389/fams.2021.702486

1 INTRODUCTION

An important problem in many applications is the identification of a function from measurements or random samples. For this problem to be well-posed, some prior information about the function has to be assumed and a common requirement is that the function can be approximated in a finite dimensional ansatz space. For the purpose of extracting governing equations the most famous approach in recent years has been SINDy [1]. However, the applicability of SINDy to high-dimensional problems is limited since truly high-dimensional problems require a nonlinear parameterization of the ansatz space. One particular reparametrization that has proven itself in many applications are tensor networks. These allow for a straight-forward extension of SINDy [2] but can also encode additional structure as presented in [3]. The compressive capabilities of tensor networks originate from this ability to exploit additional structure like smoothness, locality or self-similarity and have hence been used in solving high-dimensional equations [4–7]. In the context of optimal control tensor train networks have been utilized for solving the Hamilton–Jacobi–Bellman equation in [8,9], for solving backward stochastic differential equations in [10] and for the calculation of stock options prices in [11,12]. In the context of uncertainty quantification they are used in [13–15] and in the context of image classification they are used in [16,17].

A common thread in these publications is the parametrization of a high-dimensional ansatz space by a tensor train network which is then optimized. In most cases this means that the least-squares error of the parametrized function to the data is minimized. There exist many methods to perform this minimization. A well-known algorithm in the mathematics community is the *alternating least-squares (ALS)* [18,19], which is related to the famous DMRG method [20] for solving the Schrödinger equation in quantum physics. Although, not directly suitable for recovery tasks, it became apparent that DMRG and ALS can be adapted to work in this context. Two of these extensions to the ALS algorithm are the *stablilized ALS approximation (SALSA)* [21] and the *block alternating steepest descent for Recovery (bASD)* algorithm [13]. Both adapt the tensor network ranks and are better suited to the problem of data identification. Since the set of tensor trains of fixed rank forms a manifold [22] it is also possible to perform gradient based optimization schemes [48]. This however is not a path that we pursue in this work. Our contribution extends the ALS (and SALSA) algorithm and we believe that it can be applied to many of the fields stated above.

In this work we consider ansatz spaces of homogeneous polynomials of fixed degree and their extension to polynomials of bounded degree. We introduce the concept of block-sparsity as an efficient way to parametrize homogeneous polynomials with low rank tensors. Although, this is not the first instance in which sparsity is used in the context of low-rank tensors (see [24–26]), we believe, that this is the first time where block-sparsity is used to parametrize homogeneous polynomials. The sparsity used in the previous works is substantially different to the block-sparsity discussed in this work. Block-sparsity is preserved under most tensor network operations such as summation, orthogonalization and rounding and the parametrization of tangent spaces which is not the case for standard sparsity. This is important since orthogonalization is an essential part of numerically stable and efficient optimization schemes and means that most of the existing tensor methods, like HSVD (see [27]), ALS, SALSA or Riemannian optimization can be performed in this format. We also show that, if the symmetric tensor of a homogeneous polynomial is banded, it can be represented very efficiently in the tensor train format, since the sizes of the non-zero blocks can be bounded independently of the number of variables. In physics this property can be associated with the property of locality, which can be used to identify cases where tensor trains work exceptionally well.

Quantum physicists have used the concept of block-sparsity for at least a decade [28] but it was introduced to the mathematics community only recently in [29]. In the language of quantum mechanics one would say that there exists an operator for which the coefficient tensor of any homogeneous polynomial is an eigenvector. This encodes a symmetry, where the eigenvalue of this eigenvector is the degree of the homogeneous polynomial, which acts as a quantum number and corresponds to the particle number of bosons and fermions.

The presented approach is very versatile and can be combined with many polynomial approximation strategies like the use of Taylor's theorem in [30] and there exist many approximation theoretic results that ensure a good approximation with a low degree polynomial for many classes of functions (see e.g. [31]).

In addition to the approximation theoretic results, we can motivate these polynomial spaces by thinking about the sample complexity for successful recovery in the case of regression problems. In [32] it was shown that for tensor networks the sample complexity, meaning the number of data points needed, is related to the dimension of the high-dimensional ansatz space. But, these huge sample sizes are not needed in most practical examples [14]. This suggests that the regularity of the sought function must have a strong influence on the number of samples that are required. However, for most practical applications, suitable regularity guarantees cannot be made — neither for the best approximation nor for the initial guess, nor any iterate of the optimization process. By restricting ourselves to spaces of homogeneous polynomials, the gap between observed sample complexity and proven worst-case bound is reduced.

In the regression setting, this means that we kill two birds with one stone. By applying block-sparsity to the coefficient tensor we can restrict the ansatz space to well-behaved functions which can be identified with a reasonable sample size. At the same time we reduce the number of parameters and speed up the least-squares minimization task. Finally, note that this parametrization allows practitioners to devise algorithms that are adaptive in the degree of the polynomial, thereby increasing the computational resource utilization even further. This solves a real problem in practical applications where the additional and unnecessary degrees of freedom of conventional low-rank tensor formats cause many optimization algorithms to get stuck in local minima.

The remainder of this work is structured as follows. *Notation* introduces basic tensor notation, the different parametrizations of polynomials that are used in this work and then formulates the associated least-squares problems. In *Theoretical Foundation* we state the known results on sampling complexity and block sparsity. Furthermore, we set the two results in relation and argue why this leads to more favorable ansatz spaces. This includes a proof of rank-bounds for a class of homogeneous polynomials which can be represented particularly efficient as tensor trains. *Method Description* derives two parametrizations from the results of *Theoretical Foundation* and presents the algorithms that are used to solve the associated least-squares problems. Finally, *Numerical Results* gives some numerical results for different classes of problems focusing on the comparison of the sample complexity for the full- and sub-spaces. Most notably, the recovery of a quantity of interest for a parametric PDE, where our approach achieves successful recovery with relatively few parameters and samples. We observed that for suitable problems the number of parameters can be reduced by a factor of almost 10.

2 NOTATION

In our opinion, using a graphical notation for the involved contractions in a tensor network drastically simplifies the expressions making the whole setup more approachable. This section introduces this graphical notation for tensor networks, the spaces that will be used in the remainder of this work and the regression framework.

2.1 Tensors and Indices

Definition 2.1. Let $d \in \mathbb{N}$. Then $\boldsymbol{n} = (n_1, \ldots, n_d) \in \mathbb{N}^d$ is called a *dimension tuple of order d* and $x \in \mathbb{R}^{n_1 \times \cdots \times n_d} =: \mathbb{R}^{\boldsymbol{n}}$ is called a *tensor of order d and dimension* $\boldsymbol{n}$. Let $\mathbb{N}_n = \{1, \ldots, n\}$ then a tuple $(l_1, \ldots, l_d) \in \mathbb{N}_{n_1} \times \cdots \times \mathbb{N}_{n_d} = : \mathbb{N}_{\boldsymbol{n}}$ is called a *multi-index* and the corresponding entry of x is denoted by $x\,(l_1, \ldots, l_d)$. The positions $1, \ldots, d$ of the indices $l_1, \ldots, l_d$ in the expression $x\,(l_1, \ldots, l_d)$ are called *modes of x*.

To define further operations on tensors it is often useful to associate each mode with a symbolic index.

Definition 2.2. A *symbolic index i* of dimension n is a placeholder for an arbitrary but fixed natural number between 1 and n. For a dimension tuple $\boldsymbol{n}$ of order d and a tensor $x \in \mathbb{R}^{\boldsymbol{n}}$ we may write $x\,(i_1, \ldots, i_d)$ and tacitly assume that i_k are indices of dimension n_k for each $k = 1, \ldots, d$. When standing for itself this notation means $x(i_1, \ldots, i_d) = x \in \mathbb{R}^{\boldsymbol{n}}$ and may be used to *slice* the tensor

$$x\,(i_1, l_2, \ldots, l_d) \in \mathbb{R}^{n_1}$$

where $l_k \in \mathbb{N}_{n_k}$ are fixed indices for all $k = 2, \ldots, d$. For any dimension tuple $\boldsymbol{n}$ of order d we define the symbolic multi-index $i^{\boldsymbol{n}} = (i_1, \ldots, i_d)$ of dimension $\boldsymbol{n}$ where i_k is a symbolic index of dimension n_k for all $k = 1, \ldots, d$.

Remark 2.3. We use the letters i and j (with appropriate subscripts) for symbolic indices while reserving the letters k, l and m for ordinary indices.

Example 2.4. Let x be an order 2 tensor with mode dimensions n_1 and n_2, i.e. an n_1-by-n_2 matrix. Then $x\,(\ell_1, j)$ denotes the ℓ_1-th row of x and $x\,(i, \ell_2)$ denotes the ℓ_2-th column of x.

Inspired by Einstein notation we use the concept of symbolic indices to define different operations on tensors.

Definition 2.5. Let i_1 and i_2 be (symbolic) indices of dimension n_1 and n_2, respectively and let φ be a bijection

$$\varphi\colon \mathbb{N}_{n_1} \times \mathbb{N}_{n_2} \to \mathbb{N}_{n_1 n_2}.$$

We then define *the product of indices* with respect to φ as $j = \varphi(i_1, i_2)$ where j is a (symbolic) index of dimension $n_1 n_2$. In most cases the choice of bijection is not important and we will write $i_1 \cdot i_2 {:=} \varphi(i_1, i_2)$ for an arbitrary but fixed bijection φ. For a tensor x of dimension (n_1, n_2) the expression

$$y\,(i_1 \cdot i_2) = x\,(i_1, i_2)$$

defines the tensor y of dimension $(n_1 n_2)$ while the expression

$$x\,(i_1, i_2) = y\,(i_1 \cdot i_2)$$

defines $x \in \mathbb{R}^{n_1 \times n_2}$ from $y \in \mathbb{R}^{n_1 n_2}$.

Definition 2.6. Consider the tensors $x \in \mathbb{R}^{\boldsymbol{n}_1 \times a \times \boldsymbol{n}_2}$ and $y \in \mathbb{R}^{\boldsymbol{n}_3 \times b \times \boldsymbol{n}_4}$. Then the expression

$$z\left(i^{\boldsymbol{n}_1}, i^{\boldsymbol{n}_2}, j_1, j_2, i^{\boldsymbol{n}_3}, i^{\boldsymbol{n}_4}\right) = x\left(i^{\boldsymbol{n}_1}, j_1, i^{\boldsymbol{n}_2}\right) \cdot y\left(i^{\boldsymbol{n}_3}, j_2, i^{\boldsymbol{n}_4}\right) \qquad (1)$$

defines the tensor $z \in \mathbb{R}^{\boldsymbol{n}_1 \times \boldsymbol{n}_2 \times a \times b \times \boldsymbol{n}_3 \times \boldsymbol{n}_4}$ in the obvious way. Similary, for $a = b$ the expression

$$z\left(i^{\boldsymbol{n}_1}, i^{\boldsymbol{n}_2}, j, i^{\boldsymbol{n}_3}, i^{\boldsymbol{n}_4}\right) = x\left(i^{\boldsymbol{n}_1}, j, i^{\boldsymbol{n}_2}\right) \cdot y\left(i^{\boldsymbol{n}_3}, j, i^{\boldsymbol{n}_4}\right) \qquad (2)$$

defines the tensor $z \in \mathbb{R}^{\boldsymbol{n}_1 \times \boldsymbol{n}_2 \times a \times \boldsymbol{n}_3 \times \boldsymbol{n}_4}$. Finally, also for $a = b$ the expression

$$z\left(i^{\boldsymbol{n}_1}, i^{\boldsymbol{n}_2}, i^{\boldsymbol{n}_3}, i^{\boldsymbol{n}_4}\right) = x\left(i^{\boldsymbol{n}_1}, j, i^{\boldsymbol{n}_2}\right) \cdot y\left(i^{\boldsymbol{n}_3}, j, i^{\boldsymbol{n}_4}\right) \qquad (3)$$

defines the tensor $z \in \mathbb{R}^{\boldsymbol{n}_1 \times \boldsymbol{n}_2 \times \boldsymbol{n}_3 \times \boldsymbol{n}_4}$ as

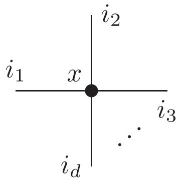
$$z\left(i^{\boldsymbol{n}_1}, i^{\boldsymbol{n}_2}, i^{\boldsymbol{n}_3}, i^{\boldsymbol{n}_4}\right) = \sum_{k=1}^{a} x\left(i^{\boldsymbol{n}_1}, k, i^{\boldsymbol{n}_2}\right) \cdot y\left(i^{\boldsymbol{n}_3}, k, i^{\boldsymbol{n}_4}\right).$$

We choose this description mainly because of its simplicity and how it relates to the implementation of these operations in the numeric libraries numpy [33] and xerus [34].

2.2 Graphical Notation and Tensor Networks

This section will introduce the concept of *tensor networks* [35] and a graphical notation for certain operations which will simplify working with these structures. To this end we reformulate the operations introduced in the last section in terms of nodes, edges and half-edges.

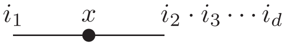
Definition 2.7. For a dimension tuple $\boldsymbol{n}$ of order d and a tensor $x \in \mathbb{R}^{\boldsymbol{n}}$ the *graphical representation* of x is given by.



where the node represents the tensor and the half-edges represent the d different modes of the tensor illustrated by the symbolic indices $i_1, \ldots, i_d$.

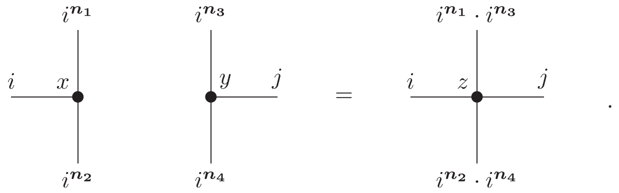
With this definition we can write the reshapings of Defintion 2.5 simply as

$x(i_1, i_2 \cdot i_3 \cdots i_d) \quad =$

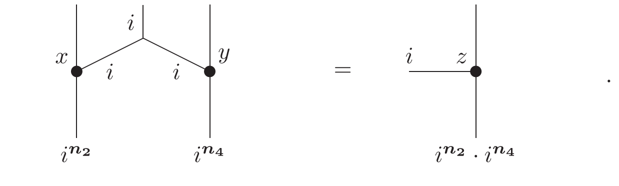


and also simplify the binary operations of Definition 2.6.

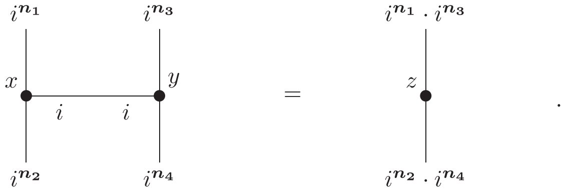
Definition 2.8. Let $x \in \mathbb{R}^{\boldsymbol{n}_1 \times a \times \boldsymbol{n}_2}$ and $y \in \mathbb{R}^{\boldsymbol{n}_3 \times b \times \boldsymbol{n}_4}$ be two tensors. Then Operation **Eq. 1** is represented by



and defines $z \in \mathbb{R}^{\cdots \times a \times b \times \cdots}$. For $a = b$ Operation **Eq. 2** is represented by



and defines $z \in \mathbb{R}^{\cdots\times a\times\cdots}$ and Operation **Eq. 3** defines $z \in \mathbb{R}^{\cdots\times\cdots}$ by.



With these definitions we can compose entire networks of multiple tensors which are called tensor networks.

2.3 The Tensor Train Format

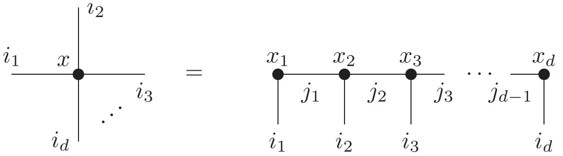
A prominent example of a tensor network is the *tensor train (TT)* [19,36], which is the main tensor network used throughout this work. This network is discussed in the following subsection.

Definition 2.9. Let $\boldsymbol{n}$ be an dimensional tuple of order-d. The TT format decomposes an order d tensor $x \in \mathbb{R}^{\boldsymbol{n}}$ into d *component tensors* $x_k \in \mathbb{R}^{r_{k-1}\times n_k\times r_k}$ for $k = 1, \ldots, d$ with $r_0 = r_d = 1$. This can be written in tensor network formula notation as

$$x(i_1, \cdots, i_d) = x_1(i_1, j_1)\cdot x_2(j_1, i_2, j_2)\cdots x_d(j_{d-1}, i_d).$$

The tuple $(r_1, \ldots, r_{d-1})$ is called the *representation rank* of this representation.

In graphical notation it looks like this.



Remark 2.10. Note that this representation is not unique. For any pair of matrices (A, B) that satisfies $AB = Id$ we can replace x_k by $x_k(i_1, i_2, j)\cdot A(j, i_3)$ and x_{k+1} by $B(i_1, j)\cdot x(j, i_2, i_3)$ without changing the tensor x.

The representation rank of x is therefore dependent on the specific representation of x as a TT, hence the name. Analogous to the concept of matrix rank we can define a minimal necessary rank that is required to represent a tensor x in the TT format.

Definition 2.11. The *tensor train rank* of a tensor $x \in \mathbb{R}^{\boldsymbol{n}}$ with tensor train components $x_1 \in \mathbb{R}^{n_1\times r_1}$, $x_k \in \mathbb{R}^{r_{k-1}\times n_k\times r_k}$ for $k = 2, \ldots, d-1$ and $x_d \in \mathbb{R}^{r_{d-1}\times n_d}$ is the set

$$\mathrm{TT}-\mathrm{rank}(x) = (r_1, \ldots, r_d)$$

of minimal r_k's such that the x_k compose x.

In [[22], Theorem 1a] it is shown that the TT-rank can be computed by simple matrix operations. Namely, r_k can be computed by joining the first k indices and the remaining $d-k$ indices and computing the rank of the resulting matrix. At last, we need to introduce the concept of left and right orthogonality for the tensor train format.

Definition 2.12. Let $x \in \mathbb{R}^{\boldsymbol{m}\times\boldsymbol{n}}$ be a tensor of order $d+1$. We call x *left orthogonal* if

$$x(i^{\boldsymbol{m}}, j_1)\cdot x(i^{\boldsymbol{m}}, j_2) = Id(j_1, j_2).$$

Similarly, we call a tensor $x \in \mathbb{R}^{m\times\boldsymbol{n}}$ of order $d+1$ *right orthogonal* if

$$x(i_1, j^{\boldsymbol{n}})\cdot x(i_2, j^{\boldsymbol{n}}) = Id(i_1, i_2).$$

A tensor train is *left orthogonal* if all component tensors $x_1, \ldots, x_{d-1}$ are left orthogonal. It is *right orthogonal* if all component tensors $x_2, \ldots, x_d$ are right orthogonal.

Lemma 2.1 [36]**.** *For every tensor $x \in \mathbb{R}^{\boldsymbol{n}}$ of order d we can find left and right orthogonal decompositions.*

For technical purposes it is also useful to define the so-called *interface tensors*, which are based on left and right orthogonal decompositions.

Definition 2.13. Let x be a tensor train of order d with rank tuple $\boldsymbol{r}$.

For every $k = 1, \ldots, d$ and $\ell = 1, \ldots, r_k$, the ℓ-th *left interface vector* is given by

$$\tau^{\leq}_{k,\ell}(x)(i_1, i_2, \cdots, i_k) = x_1(i_1, j_1)\cdots x_k(j_{k-1}, i_k, \ell)$$

where x is assumed to be left orthogonal. The ℓ-th *right interface vector* is given by

$$\tau^{\geq}_{k+1,\ell}(x)(i_{k+1}, \cdots, i_d) = x_{k+1}(\ell, i_{k+1}, j_{k+1})\cdots x_d(j_{d-1}, i_d)$$

where x is assumed to be right orthogonal.

2.4 Sets of Polynomials

In this section we specify the setup for our method and define the majority of the different sets of polynomials that are used. We start by defining dictionaries of one dimensional functions which we then use to construct the different sets of high-dimensional functions.

Definition 2.14. Let $p \in \mathbb{N}$ be given. A function dictionary of size p is a vector valued function $\Psi = (\Psi_1, \ldots, \Psi_p)\colon \mathbb{R}\rightarrow\mathbb{R}^p$.

Example 2.15. Two simple examples of a function dictionary that we use in this work are given by the monomial basis of dimension p, i.e.

$$\Psi_{\text{monomial}}(x) = \begin{pmatrix} 1 & x & x^2 & \ldots & x^{p-1} \end{pmatrix}^T \tag{4}$$

and by the basis of the first p Legendre polynomials, i.e.

$$\Psi_{\text{Legendre}}(x) = \begin{pmatrix} 1 & x & \frac{1}{2}(3x^2-1) & \frac{1}{2}(5x^3-3x) & \ldots \end{pmatrix}^T. \tag{5}$$

Using function dictionaries we can define the following high-dimensional space of multivariate functions. Let Ψ be a function dictionary of size $p \in \mathbb{N}$. The d-th order product space that corresponds to the function dictionary Ψ is the linear span

$$V^d_p := \langle \bigotimes_{k=1}^{d} \Psi_{m_k} :\ \boldsymbol{m} \in \mathbb{N}^d_p \rangle. \tag{6}$$

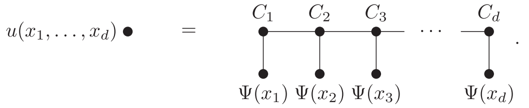
This means that every function $u \in V^d_p$ can be written as

$$u(x_1, \ldots, x_d) = c(i_1, \ldots, i_d)\prod_{k=1}^{d}\Psi(x_k)(i_k) \tag{7}$$

with a coefficient tensor $c \in \mathbb{R}^{\boldsymbol{p}}$ where $\boldsymbol{p} = (p, \ldots, p)$ is a dimension tuple of order d. Note that equation **Eq. 7** uses the index notation from Definition 2.6 with arbitrary but fixed x_k's. Since $\mathbb{R}^{\boldsymbol{p}}$ is an intractably large space, it makes sense for numerical purposes to consider the subset

$$T_r\left(V_p^d\right) := \left\{u \in V_p^d : \ \mathrm{TT-rank}(c) \leq r\right\} \tag{8}$$

where the TT rank of the coefficient is bounded. Every $u \in T_r(V_p^d)$ can thus be represented graphically as



where the C_k's are the components of the tensor train representation of the coefficient tensor $c \in \mathbb{R}^{\boldsymbol{p}}$ of $u \in V_p^d$.

Remark 2.16. In this way every tensor $c \in \mathbb{R}^{\boldsymbol{p}}$ (in the tensor train format) corresponds one to one to a function $u \in V_p^d$.

An important subspace of V_p^d is the space of homogeneous polynomials. For the purpose of this paper we define the subspace of homogeneous polynomials of degree g as the space

$$W_g^d := \langle \otimes_{k=1}^d \Psi_{m_k} : \ \boldsymbol{m} \in \mathbb{N}_p^d \quad \text{and} \quad \sum_{k=1}^d m_k = d + g \rangle, \tag{10}$$

where again $\langle \bullet \rangle$ is the linear span. From this definition it is easy to see that a homogeneous polynomial of degree g can be represented as an element of V_p^d where the coefficient tensor c satisfies

$$c(m_1,\dots,m_d) = 0, \quad \textit{if} \quad \sum_{k=1}^d m_k \neq d + g.$$

In *Theoretical Foundation* we will introduce an efficient representation of such coefficient tensors c in a block sparse tensor format.

Using W_g^d we can also define the space of polynomials of degree at most g by

$$S_g^d = \oplus_{\tilde{g}=0}^g W_{\tilde{g}}^d. \tag{11}$$

Based on this characterization we will define a block-sparse tensor train version of this space in *Theoretical Foundation.*

2.5 Parametrizing Homogeneous Polynomials by Symmetric Tensors

In algebraic geometry the space W_g^d is considered classically only for the dictionary Ψ_{monomial} of monomials and is typically parameterized by a symmetric tensor

$$u(x) = B\left(i_1,\cdots,i_g\right)\cdot x(i_1)\cdots x\left(i_g\right), \quad x \in \mathbb{R}^d \tag{12}$$

where $\boldsymbol{d} = (d,\dots,d)$ is a dimension tuple of order g and $B \in \mathbb{R}^{\boldsymbol{d}}$ satisfies $B(m_1,\dots,m_g) = B(\sigma(m_1,\dots,m_g))$ for every permutation σ in the symmetric group S_g. We conclude this section by showing how the representation **Eq. 7** can be calculated from the symmetric tensor representation **Eq. 12**, and vice versa. By equating coefficients we find that for every $(m_1,\dots,m_d) \in \mathbb{N}_p^d$ either $m_1 + \cdots + m_d \neq d + g$ and $c(m_1,\dots,m_d) = 0$ or

$$\begin{aligned} c(m_1,\dots,m_d) &= \sum_{\{\sigma(\boldsymbol{n}) : \ \sigma \in S_g\}} B\left(\sigma\left(n_1,\dots,n_g\right)\right) \quad \text{where} \quad \left(n_1,\dots,n_g\right) \\ &= \big(\underbrace{1,\dots,1}_{m_1-1 \text{ times}},\underbrace{2,\dots,2}_{m_2-1 \text{ times}},\dots,\big) \in \mathbb{N}_d^g. \end{aligned}$$

Since B is symmetric the sum simplifies to

$$\sum_{\{\sigma(\boldsymbol{n}) : \ \sigma \in S_g\}} B\left(\sigma\left(n_1,\dots,n_g\right)\right) = \binom{g}{m_1 - 1,\dots,m_d - 1} B\left(n_1,\dots,n_g\right).$$

From this follows that for $(n_1,\dots,n_g) \in \mathbb{N}_d^g$

$$B\left(n_1,\dots,n_g\right) = \frac{1}{\binom{g}{m_1 - 1,\dots,m_d - 1}} c(m_1,\dots,m_d) \quad \text{where}$$

$$m_k = 1 + \sum_{\ell=1}^g \delta_{k,n_\ell} \quad \text{for all} \quad k = 1,\dots,d$$

and $\delta_{k,\ell}$ denotes the *Kronecker delta*. This demonstrates how our approach can alleviate the difficulties that arise when symmetric tensors are represented in the hierarchical tucker format [37] in a very simple fashion.

2.6 Least Squares

Let in the following V_p^d be the product space of a function dictionary Ψ such that $V_p^d \subseteq L_2(\Omega)$. Consider a high-dimensional function $f \in L_2(\Omega)$ on some domain $\Omega \subset \mathbb{R}^d$ and assume that the point-wise evaluation $f(x)$ is well-defined for $x \in \Omega$. In practice it is often possible to choose Ω as a product domain $\Omega = \Omega_1 \times \Omega_2 \times \cdots \Omega_d$ by extending f accordingly. To find the best approximation u_W of f in the space $W \subseteq V_p^d$ we then need to solve the problem

$$u_W = \mathrm{argmin}_{u \in W} \|f - u\|_{L_2(\Omega)}^2. \tag{13}$$

A practical problem that often arises when computing u_W is that computing the $L_2(\Omega)$-norm is intractable for large d. Instead of using classical quadrature rules one often resorts to a Monte Carlo estimation of the high-dimensional integral. This means one draws M random samples $\{x^{(m)}\}_{m=1,\dots,M}$ from Ω and estimates

$$\|f - u\|_{L_2(\Omega)}^2 \approx \frac{1}{M} \sum_{m=1}^M \|f\left(x^{(m)}\right) - u\left(x^{(m)}\right)\|_{\mathrm{F}}^2,$$

where $\|\cdot\|_F$ is the Frobenius norm. With this approximation we can define an empirical version of u_W as

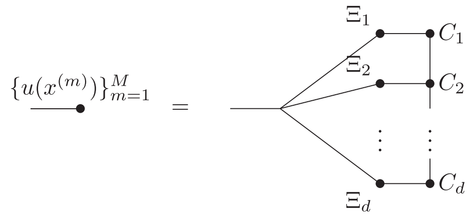
$$u_{W,M} = \underset{u \in W}{\mathrm{argmin}} \frac{1}{M} \sum_{m=1}^M \|f\left(x^{(m)}\right) - u\left(x^{(m)}\right)\|_{\mathrm{F}}^2. \tag{14}$$

For a linear space W, computing $u_{W,M}$ amounts to solving a linear system and does not pose an algorithmic problem. We use the remainder of this section to comment on the minimization problem **Eq. 14** when a set of tensor trains is used instead.

Given samples $(x^{(m)})_{m=1,\ldots,M}$ we can evaluate $u \in V_p^d$ for each $x^{(m)} = (x_1^{(m)}, \ldots, x_d^{(m)})$ using **Eq. 7** If the coefficient tensor c of u can be represented in the TT format then we can use **Eq. 9** to perform this evaluation efficiently for all samples $(x^{(m)})_{m=1,\ldots,M}$ at once. For this we introduce for each $k = 1, \ldots, d$ the matrix

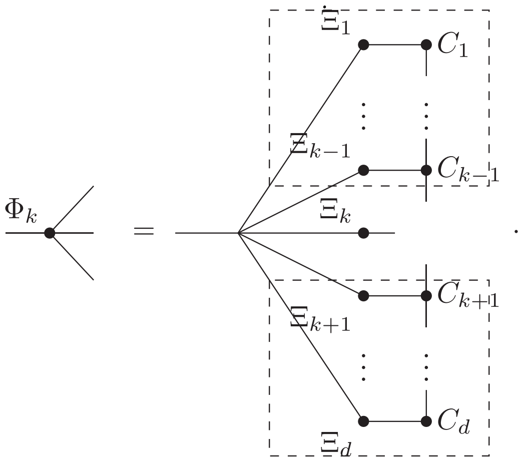
$$\Xi_k = \begin{pmatrix} \Psi\left(x_k^{(1)}\right) & \ldots & \Psi\left(x_k^{(M)}\right) \end{pmatrix} \in \mathbb{R}^{p\times M}. \tag{15}$$

Then the M-dimensional vector of evaluations of u at all given sample points is given by.



where we use Operation **Eq. 2** to join the different M-dimensional indices.

The alternating least-squares algorithm cyclically updates each component tensor C_k by minimizing the residual corresponding to this contraction. To formalize this we define the operator $\Phi_k \in \mathbb{R}^{M\times r_{k-1}\times n_k\times r_k}$ as



(16)

Then the update for C_k is given by a minimal residual solution of the linear system

$$C_k = \underset{C\in\mathbb{R}^{r_{k-1}\times n_k\times r_k}}{\operatorname{argmin}} \|\Phi_k\left(j, i_1, i_2, i_3\right)\cdot C\left(i_1, i_2, i_3\right) - F(j)\|_F^2$$

where $F(m) := y^{(m)} := f(x^{(m)})$ and i_1, i_2, i_3, j are symbolic indices of dimensions r_{k-1}, n_k, r_k, M, respectively. The particular algorithm that is used for this minimization may be adapted to the problem at hand. These contractions are the basis for our algorithms in *Method Description*. We refer to [19] for more details on the ALS algorithm.

Note that it is possible to reuse parts of the contractions in Φ_k through so called *stacks*. In this way not the entire contraction has to be computed for every k. The dashed boxes mark the parts of the contraction that can be reused. Details on that can be found in [38].

3 THEORETICAL FOUNDATION

3.1 Sample Complexity for Polynomials

The accuracy of the solution $u_{W,M}$ of **Eq. 14** in relation to u_W is subject to tremendous interest on the part of the mathematics community. Two particular papers that consider this problem are [32,39]. While the former provides sharper error bounds for the case of linear ansatz spaces the latter generalizes the work and is applicable to tensor network spaces. We now recall the relevant result for convenience.

Proposition 3.1. *For any set* $W \subseteq L^2(\Omega)\cap L^\infty(\Omega)$, *define the* variation constant

$$K(W) := \sup_{v\in W\setminus\{0\}} \frac{\|v\|_{L^\infty(\Omega)}^2}{\|v\|_{L^2(\Omega)}^2}.$$

Let $\delta \in (0, 2^{-1/2})$. *If* W *is a subset of a finite dimensional linear space and* $k := \max\{K(\{f - u_W\}), K(\{u_W\} - W)\} < \infty$ *it holds that*

$$\mathbb{P}\left[\|f - u_{W,M}\|_{L^2(\Omega)} \leq (3 + 4\delta)\|f - u_W\|_{L^2(\Omega)}\right] \geq 1 - q$$

where q *decreases exponentially with a rate of* $\ln(q) \in \mathcal{O}(-M\delta^2 k^{-2})$.

Proof. Since $k < \infty$, Theorems 2.7 and 2.12 in [32] ensure that

$$\|f - u_{W,M}\|_{L^2(\Omega)} \leq \left(1 + 2\sqrt{\frac{1+\delta}{1-\delta}}\right)\|f - u_W\|_{L^2(\Omega)}.$$

holds with a probability of at least $1 - 2C\exp\left(-\frac{1}{2}M\delta^2 k^{-2}\right)$. The constant C is independent of M and, since W is a subset of a finite dimensional linear space, depends only polynomially on δ and k^{-1}. For $\delta \in (0, 2^{-1/2})$ it holds that $\sqrt{\frac{1+\delta}{1-\delta}} \leq 1 + 2\delta$. This concludes the proof.

Note that the value of k depends only on f and on the set W but not on the particular choice of representation of W. However, the variation constant of spaces like V_p^d still depends on the underlying dictionary Ψ. Although the proposition indicates that a low value of k is necessary to achieve a fast convergence, the tensor product spaces V_p^d considered thus far does not exhibit a small variation constant. The consequence of Proposition 3.1 is that elements of this space are hard to learn in general and may require an infeasible number of samples. To see this consider $\Omega = [-1,1]^d$ and the function dictionary Ψ_{Legendre} of Legendre polynomials **Eq. 5**. Let $L \subseteq \mathbb{N}_p^d$ and define $P_{\ell}(x) := \prod_{k=1}^d \sqrt{2\ell_k - 1}(\Psi_{\text{Legendre}}(x_k))_{\ell_k}$ for all $\ell \in L$. Then, $\{P_\ell\}_{\ell\in L}$ is an L^2-orthonormal basis for the linear subspace $V := \langle P_\ell : \ell \in L\rangle \subseteq V_p^d$ and one can show that

$$K(V) = \sup_{x\in\Omega} \sum_{\ell\in L} P_\ell(x)^2 = \sum_{\ell\in L}\prod_{k=1}^d (2\ell_k - 1), \tag{17}$$

by using techniques from [[32], *Sample Complexity for Polynomials*] and the fact that each P_ℓ attains its maximum at 1. If $L = \mathbb{N}_p^d$, we can interchange the sum and product in **Eq. 17** and can conclude that $K(V_p^d) = p^{2d}$. This means that we have to restrict the space V_p^d to obtain an admissible variation constant. We propose to use the space W_g^d of homogeneous polynomials of degree g. Employing **Eq. 17** with $L = \{\boldsymbol{\ell} : |\boldsymbol{\ell}| = d + g\}$ we obtain the upper bound

$$\begin{aligned} K\left(W_g^d\right) &\leq \binom{d-1+g}{d-1} \max_{|\boldsymbol{\ell}|=d+g} \prod_{k=1}^{d} (2\ell_k - 1) \\ &\leq \binom{d-1+g}{d-1} \left(2\left\lfloor\frac{g}{d}\right\rfloor + 3\right)^{g \bmod d} \left(2\left\lfloor\frac{g}{d}\right\rfloor + 1\right)^{d-g \bmod d} \end{aligned}$$

where the maximum is estimated by observing that $(2\,(\ell_1 + 1) - 1)\,(2\ell_2 - 1) \leq (2\ell_1 - 1)\,(2\,(\ell_2 + 1) - 1) \Leftrightarrow \ell_2 \leq \ell_1$. For $g \leq d$ this results in the simplified bound $K(W_g^d) \leq (3\mathrm{e}\frac{d-1+g}{g})^g$, where e is the Euler number. This improves the variation constant substantially compared to the bound $K(V_p^d) \leq p^{2d}$, when $g \ll d$. A similar bound for the dictionary of monomials Ψ_{monomial} is more involved but can theoretically be computed in the same way.

In this work, we focus on the case where the samples are drawn according to a probability measure on Ω. This however is not a necessity and it is indeed beneficial to draw the samples from an adapted sampling measure. Doing so, the theory in [32] ensures that $K(V) = \dim(V)$ for all linear spaces V — independent of the underlying dictionary Ψ. This in turn leads to the bounds $K(V_p^d) = p^d$ and $K(W_g^d) = \binom{d-1+g}{d-1} \leq \left(\mathrm{e}\frac{d-1+g}{g}\right)g$ for $g \leq d$. These optimally weighted least-squares methods however, are not the focus of this work and we refer the interested reader to the works [39,40].

3.2 Block Sparse Tensor Trains

Now that we have seen that it is advantageous to restrict ourselves to the space W_g^d we need to find a way to do so without loosing the advantages of the tensor train format. In [29] it was rediscovered from the physics community (see [28]) that if a tensor train is an eigenvector of certain Laplace-like operators it admits a block sparse structure. This means for a tensor train c the components C_k have zero blocks. Furthermore, this block sparse structure is preserved under key operations, like e.g. the TT-SVD. One possible operator which introduces such a structure is the Laplace-like operator

$$L = \sum_{k=1}^{d} \left(\bigotimes_{\ell=1}^{k-1} I_p\right) \otimes \operatorname{diag}(0, 1, \ldots, p-1) \otimes \left(\bigotimes_{\ell=k+1}^{d} I_p\right). \tag{18}$$

This is the operator mentioned in the introduction encoding a quantum symmetry. In the context of quantum mechanics this operator is known as the bosonic particle number operator but we simply call it the degree operator. The reason for this is that for the function dictionary of monomials Ψ_{monomial} the eigenspaces of L for eigenvalue g are associated with homogeneous polynomials of degreee g. Simply put, if the coefficient tensor c for the multivariate polynomial $u \in V_p^d$ is an eigenvector of L with eigenvalue g, then u is homogeneous and the degree of u is g. In general there are polynomials in V_p^d with degree up to $(p - 1)d$. To state the results on the block-sparse representation of the coefficient tensor we need the partial operators

$$\begin{aligned} L_k^{\leq} &= \sum_{m=1}^{k} \left(\bigotimes_{\ell=1}^{m-1} I_p\right) \otimes \operatorname{diag}(0, 1, \ldots, p-1) \otimes \left(\bigotimes_{\ell=m+1}^{k} I_p\right) \\ L_{k+1}^{\geq} &= \sum_{m=k+1}^{d} \left(\bigotimes_{\ell=k+1}^{m-1} I_p\right) \otimes \operatorname{diag}(0, 1, \ldots, p-1) \otimes \left(\bigotimes_{\ell=m+1}^{d} I_p\right), \end{aligned}$$

for which we have

$$L = L_k^{\leq} \otimes \bigotimes_{\ell=k+1}^{d} I_p + \bigotimes_{l=1}^{k} I_p \otimes L_{k+1}^{\geq}.$$

In the following we adopt the notation $x = Lc$ to abbreviate the equation

$$x(i_1, \ldots, i_d) = L(i_1, \ldots, i_d, j_1, \ldots, j_d)c(j_1, \ldots, j_d)$$

where L is a tensor operator acting on a tensor c with result x.

Recall that by Remark 2.16 every TT corresponds to a polynomial by multiplying function dictionaries onto the cores. This means that for every $\ell = 1, \ldots, r$ the TT $\tau_{k,\ell}^{\leq}(c)$ corresponds to a polynomial in the variables $x_1, \ldots, x_k$ and the TT $\tau_{k+1,\ell}^{\geq}(c)$ corresponds to a polynomial in the variables $x_{k+1}, \ldots, x_d$. In general these polynomials are not homogeneous, i.e. they are not eigenvectors of the degree operators $L_k^{\leq}$ and $L_{k+1}^{\geq}$. But since TTs are not uniquely defined (cf. Remark 2.10) it is possible to find transformations of the component tensors C_k and C_{k+1} that do not change the tensor c or the rank r but result in a representation where each $\tau_{k,\ell}^{\leq}(c)$ and each $\tau_{k+1,\ell}^{\geq}(c)$ correspond to a homogeneous polynomial. Thus, if c represents a homogeneous polynomial of degree g and $\tau_{k,\ell}^{\leq}(c)$ is homogeneous with $\deg(\tau_{k,\ell}^{\leq}(c)) = \tilde{g}$ then $\tau_{k+1,\ell}^{\geq}(c)$ must be homogeneous with $\deg(\tau_{k,\ell}^{\geq}(c)) = g - \tilde{g}$.

This is put rigorously in the first assertion in the subsequent Theorem 3.2. There $\mathcal{S}_{k,\tilde{g}}$ contains all the indices ℓ for which the reduced basis polynomials satisfy $\deg(\tau_{k,\ell}^{\leq}(c)) = \tilde{g}$. Equivalently, it groups the basis functions $\tau_{k+1,\ell}^{\geq}(c)$ into functions of order $g - \tilde{g}$. The second assertion in Theorem 3.2 states that we can only obtain a homogeneous polynomial of degree $\tilde{g} + m$ in the variables $x_1, \ldots, x_k$ by multiplying a homogeneous polynomial of degree $\tilde{g}$ in the variables $x_1, \ldots, x_{k-1}$ with a univariate polynomial of degree m in the variable x_k. This provides a constructive argument for the proof and can be used to ensure block-sparsity in the implementation. Note that this condition forces entire blocks in the component tensor C_k in equation **(20)** to be zero and thus decreases the degrees of freedom.

Theorem 3.2 [[29], *Theorem 3.2]. Let* $\boldsymbol{p} = (p, \ldots, p)$ *be a dimension tuple of size d and* $c \in \mathbb{R}^{\boldsymbol{p}}\ \{0\}$, *be a tensor train of rank* $r = (r_1, \ldots, r_{d-1})$. *Then* $Lc = gc$ *if and only if c has a representation with component tensors* $C_k \in \mathbb{R}^{r_{k-1} \times p \times r_k}$ *that satisfies the following two properties.*

1. For all $\tilde{g} \in \{0, 1, \ldots, g\}$ *there exist* $\mathcal{S}_{k,\tilde{g}} \subseteq \{1, \ldots, r_k\}$ *such that the left and right unfoldings satsify*

$$\begin{aligned} L_k^{\leq} \tau_{k,\ell}^{\leq}(c) &= \tilde{g} \tau_{k,\ell}^{\leq}(c) \\ L_{k+1}^{\geq} \tau_{k+1,\ell}^{\geq}(c) &= (g - \tilde{g}) \tau_{k+1,\ell}^{\geq}(c) \end{aligned} \tag{19}$$

for $\ell \in \mathcal{S}_{k,\tilde{g}}$.

2. The component tensors satisfy a block structure in the sets $\mathcal{S}_{k,\tilde{g}}$ *for* $m = 1, \ldots, p$

$$\begin{aligned} C_k(\ell_1, m, \ell_2) \neq 0 \quad \Rightarrow \quad & \exists\ 0 \leq \tilde{g} \leq g - (m-1): \\ & \ell_1 \in \mathcal{S}_{k-1,\tilde{g}} \wedge \ell_2 \in \mathcal{S}_{k,\tilde{g}+(m-1)} \end{aligned} \tag{20}$$

where we set $\mathcal{S}_{0,0} = \mathcal{S}_{d,g} = \{1\}$.

Note that this generalizes to other dictionaries and is not restricted to monomials.

Although, block sparsity also appears for $g + 1 \neq p$ we restrict ourselves to the case $g + 1 = p$ in this work. Note that then the eigenspace of L for the eigenvalue g has a dimension equal to the dimension of the space of homogeneous polynomials, namely $\binom{d-1+g}{d-1}$. Defining $\rho_{k,\tilde{g}} := |\mathcal{S}_{k,\tilde{g}}|$, we can derive the following rank bounds.

Lemma 3.3 [[29], *Lemma 3.6]. Let* $\boldsymbol{p} = (p, \ldots, p)$ *be a dimension tuple of size* d *and* $c \in \mathbb{R}^{\boldsymbol{p}}\ \{0\}$, *with* $Lc = gc$. *Assume that* $g + 1 = p$ *then the block sizes* $\rho_{k,\tilde{g}}$ *from Theorem 3.2 are bounded by*

$$\rho_{k,\,\tilde{g}} \leq \min\left\{\binom{k+\tilde{g}-1}{k-1}, \binom{d-k+g-\tilde{g}-1}{d-1}\right\} \tag{21}$$

for all $k = 1, \ldots, d-1$ *and* $\tilde{g} = 0, \ldots, g$ *and* $\rho_{k,0} = \rho_{k,g} = 1$.

The proof of this lemma is based on a simple combinatorial argument. For every k consider the size of the groups $\rho_{k-1,\bar{g}}$ for $\bar{g} \leq \tilde{g}$. Then $\rho_{k,\tilde{g}}$ can not exceed the sum of these sizes. Similarly, $\rho_{k,\bar{g}}$ can not exceed $\sum_{\bar{g}\leq\tilde{g}} \rho_{k+1,\bar{g}}$. Solving these recurrence relations yields the bound.

Example 3.1 (*Block Sparsity*)**.** Let $p = 4$ and $g = 3$ be given and let c be a tensor train such that $Lc = gc$. Then for $k = 2, \ldots, d-1$ the component tensors C_k of c exhibit the following block sparsity (up to permutation). For indices i of order r_{k-1} and j of order r_k

$$C_k(i,1,j) = \begin{pmatrix} * & 0 & 0 & 0 \\ 0 & * & 0 & 0 \\ 0 & 0 & * & 0 \\ 0 & 0 & 0 & * \end{pmatrix} \quad C_k(i,2,j) = \begin{pmatrix} 0 & * & 0 & 0 \\ 0 & 0 & * & 0 \\ 0 & 0 & 0 & * \\ 0 & 0 & 0 & 0 \end{pmatrix}$$
$$C_k(i,3,j) = \begin{pmatrix} 0 & 0 & * & 0 \\ 0 & 0 & 0 & * \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad C_k(i,4,j) = \begin{pmatrix} 0 & 0 & 0 & * \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

This block structure results from sorting the indices i and j in such a way that $\max \mathcal{S}_{k,\tilde{g}} + 1 = \min \mathcal{S}_{k,\tilde{g}+1}$ for every $\tilde{g}$.

The maximal block sizes $\rho_{k,\tilde{g}}$ for $k = 1, \ldots, d-1$ are given by

$$\rho_{k,0} = 1, \quad \rho_{k,1} = \min\{k, d-k\}, \quad \rho_{k,2} = \min\{k, d-k\}, \quad \rho_{k,3} = 1.$$

As one can see by Lemma 3.3 the block sizes $\rho_{k,\tilde{g}}$ can still be quite high.

The expressive power of tensor train parametrizations can be understood by different concepts, such as locality or self similarity. We use the remainder of this section to provide d-independent rank bounds in the context of locality.

Definition 3.2. Let $u \in W_g^d$ be a homogeneous polynomial and B be the symmetric coefficient tensor introduced in *Parametrizing homogeneous polynomials by symmetric tensors* We say that u has a variable locality of K_{loc} if $B(\ell_1, \ldots, \ell_g) = 0$ for all $(\ell_1, \ldots, \ell_g) \in \mathbb{N}_d^g$ with

$$\max\{|\ell_{m_1} - \ell_{m_2}| : \ m_1, m_2 = 1, \ldots, g\} > K_{\mathrm{loc}}.$$

Example 3.3. Let u be a homogeneous polynomial of degree 2 with variable locality K_{loc}. Then the symmetric matrix B (cf. **Eq. 12**) is K_{loc}-banded. For $K_{\mathrm{loc}} = 0$ this means that B is diagonal and that u takes the form

$$u(x) = \sum_{\ell=1}^{d} B_{\ell\ell} x_\ell^2.$$

This shows that variable locality removes mixed terms.

Remark 3.4. The locality condition in the following Theorem 3.4 is a sufficient, but in no way necessary, condition for a low rank. But since locality is a prominent feature of many physical phenomena, this condition allows us to identify an entire class of highly relevant functions which can be approximated very efficiently.

Consider, for example, a many-body system in one dimension, where each body is described by position and velocity coordinates. If the influence of neighboring bodies is much higher than the influence of more distant ones, the coefficients of the polynomial parts that depend on multiple variables often can be neglected. The forces in this system then exhibit a locality structure. An example of this is given in equation **Eq. 6** in [3], where this structure is exhibited by the force that acts on the bodies. A similar structure also appears in the microscopic traffic models in *Notation* of [41].

Another example is given by the polynomial chaos expansion of the stochastic process

$$X_t(\xi_1, \ldots, \xi_t) := c(i_1, \ldots, i_t) \prod_{k=1}^{t} \Psi(\xi_k)(i_k)$$

for $t \in \mathbb{N}$, where Ψ is the function dictionary of Hermite polynomials. In many applications, it is justified to assume that the magnitude of the covariance $\mathrm{Cov}(X_{t_1}, X_{t_2})$ decays with the distance of the indices $|t_1 - t_2|$ If the covariance decays fast enough, the coefficient tensor exhibits approximate locality, i.e. it can be well approximated by a coefficient tensor that satisfies the locality condition. Examples of this are Gaussian processes with a Matérn kernel [42,43] or Markov processes.

Theorem 3.4. *Let* $\boldsymbol{p} = (p, \ldots, p)$ *be a dimension tuple of size* d *and* $c \in \mathbb{R}^{\boldsymbol{p}}\ \{0\}$ *correspond to a homogeneous polynomial of degree* $g + 1 = p$ *(i.e.* $Lc = gc$*) with variable locality* K_{loc}. *Then the block sizes* $\rho_{k\tilde{g}}$ *are bounded by*

$$\rho_{k,\tilde{g}} \leq \sum_{\ell=1}^{K_{\mathrm{loc}}} \min\left\{\binom{K_{\mathrm{loc}} - \ell + 1 + \tilde{g} - 2}{K_{\mathrm{loc}} - \ell}, \binom{\ell + g - \tilde{g} - 2}{\ell - 1}\right\} \tag{22}$$

for all $k = 1, \ldots, d-1$ *and* $\tilde{g} = 1, \ldots, g-1$ *as well as* $\rho_{k,0} = \rho_{k,g} = 1$.

Proof. For fixed $g > 0$ and a fixed component C_k recall that for each l the tensor $\tau_{k,l}^{\leq}(c)$ corresponds to a reduced basis function v_l in the variables $x_1, \ldots, x_k$ and that for each l the tensor $\tau_{k+1,l}^{\geq}(c)$ corresponds to a reduced basis function w_l in the variables $x_{k+1}, \ldots, x_d$. Further recall that the sets $\mathcal{S}_{k,\tilde{g}}$ group these v_l and w_l. For all $l \in \mathcal{S}_{k,\tilde{g}}$ it holds that $\deg(v_l) = \tilde{g}$ and $\deg(w_l) = g - \tilde{g}$. For $\tilde{g} = 0$ and $\tilde{g} = g$ we know from Lemma 3.3 that $\rho_{k,\tilde{g}} = 1$. Now fix any $0 < \tilde{g} < g$ and arrange all the polynomials v_l of degree $\tilde{g}$ in a vector v and all polynomials w_l of degree $g - \tilde{g}$ in a vector w. Then every polynomial of the form $v^\top Qw$ for some matrix Q satisfies the degree constraint and the maximal possible rank of Q provides an upper bound for the block size $\rho_{k,\tilde{g}}$. However, due to the locality constraint we know that certain entries of Q have to be zero. We denote a variable of a polynomial as inactive if the polynomial is constant with respect to changes in this variable and active otherwise. Assume that the polynomials in v are ordered (ascendingly) according to the smallest index of their active variables and that the polynomials in w are ordered (ascendingly) according to the largest index of their active variables. With this ordering Q takes the form

$$Q = \begin{pmatrix} & & & & 0 \\ Q_1 & & & & \\ * & Q_2 & & & \\ * & * & Q_3 & & \\ \vdots & \vdots & \vdots & \ddots & \\ * & * & * & \cdots & Q_{K_{\text{loc}}} \end{pmatrix}.$$

This means that for $\ell = 1, \ldots, K_{\text{loc}}$ each block Q_ℓ matches a polynomial v_l of degree $\tilde{g}$ in the variables $x_{k-K_{\text{loc}}+\ell}, \ldots, x_k$ with a polynomial w_l of degree $g-\tilde{g}$ in the variables $x_{k+1}, \ldots, x_{k+\ell}$.

Observe that the number of rows in Q_ℓ decreases while the number columns increases with ℓ. This means that we can subdivide Q as

$$Q = \begin{pmatrix} 0 & 0 & 0 \\ Q_{\text{C}} & 0 & 0 \\ * & Q_{\text{R}} & 0 \end{pmatrix},$$

where Q_{C} contains the blocks Q_ℓ with more rows than columns (i.e. full column rank) and Q_{R} contains the blocks Q_ℓ with more columns than rows (i.e. full row rank). So Q_{C} is a tall-and-skinny matrix while Q_{R} is a short-and-wide matrix and the rank for general Q is bounded by the sum over the column sizes of the Q_ℓ in Q_{C} plus the sum over the row sizes of the Q_ℓ in Q_{R} i.e.

$$rank(Q) = \sum_{\ell=1}^{K_{\text{loc}}} rank(Q_\ell).$$

To conclude the proof it remains to compute the row and column sizes of Q_ℓ. Recall that the number of rows of Q_ℓ equals the number of polynomials u of degree $\tilde{g}$ in the variables $x_{k-K_{\text{loc}}+\ell}, \ldots, x_k$ that can be represented as $u(x_{k-K_{\text{loc}}+\ell}, \ldots, x_k) = x_{k-K_{\text{loc}}+\ell}\tilde{u}(x_{k-K_{\text{loc}}+\ell}, \ldots, x_k)$. This corresponds to all possible $\tilde{u}$ of degree $\tilde{g}-1$ in the $K_{\text{loc}} - \ell + 1$ variables $x_{k-K_{\text{loc}}+\ell}, \ldots, x_k$. This means that

$$\#rows(Q_\ell) \leq \binom{K_{\text{loc}} - \ell + 1 + \tilde{g} - 2}{K_{\text{loc}} - \ell}$$

and a similar argument yields

$$\#columns(Q_\ell) \leq \binom{\ell + g - \tilde{g} - 2}{\ell - 1}.$$

This concludes the proof.

This lemma demonstrates how the combination of the model space W_g^d with a tensor network space can reduce the space complexity by incorporating locality.

Remark 3.5. The rank bound in Theorem 3.4 is only sharp for the highest possible rank. The precise bounds can be much smaller, especially for the first and last ranks, but are quite technical to write down. For this reason, we do not provide them.

One sees that the bound only depends on g and K_{loc} and is therefore d-independent.

Remark 3.6. The rank bounds presented in this section do not only hold for the monomial dictionary Ψ_{monomial} but for all polynomial dictionaries Ψ that satisfy $\deg(\Psi_k) = k - 1$ for all $k = 1, \ldots, p$. When we speak of polynomials of degree g, we mean the space $W_g^d = \{v \in V_p^d\colon \deg(v) = g\}$. For the dictionary of monomials Ψ_{monomial} the space W_g^d contains only homogeneous polynomials in the classical sense. However, when the basis of Legendre polynomials Ψ_{Legendre} is used one obtains a space in which the functions are not homogeneous in the classical sense. Note that we use polynomials since they have been applied successfully in practice, but other function dictionaries can be used as well. Also note that the theory is much more general as shown in [29] and is not restricted to the degree counting operator.

With Theorem 3.4 one sees that tensor trains are well suited to parametrize homogeneous polynomials of fixed degree where the symmetric coefficient tensor B (cf. **Eq. 12**) is approximately banded (see also Example 3.3). This means, that there exist an K_{loc} such that the error for a best approximation of B by a tensor B with variable locality K_{loc} is small. However, K_{loc} is not known precisely in practice but can only be assumed by physical understanding of the problem at hand. Therefore, we still rely on rank adaptive schemes to find appropriate rank and block sizes. Moreover, the locality property heavily depends on the ordering of the modes. This ordering can be optimized, for example, by using entropy measures for the correlation of different modes, as it is done in quantum chemistry (cf. [[44], Remark 4.2]) or by model selection methods (cf. [25,27,45,46]).

4 METHOD DESCRIPTION

In this section we utilize the insights of *Theoretical Foundation* to refine the approximation spaces W_g^d and S_g^d and adapt the *alternating least-squares (ALS)* method to solve the related least-squares problems. First, we define the subset

$$B_\rho\left(W_g^d\right) := \Big\{u \in W_g^d : \ c \text{ is block-sparse with } \rho_{k,\tilde{g}} \leq \rho \text{ for } 0 \leq \tilde{g} \leq g, k = 1, \ldots, d\Big\} \tag{23}$$

and provide an algorithm for the related least-squares problem in Algorithm 1 which is a slightly modified version of the classical ALS [19].[1] With this definition a straight-forward application of the concept of block-sparsity to the space S_g^d is given by

$$S_{g,\rho}^d = \bigotimes_{\tilde{g}=0}^{g} B_\rho\left(W_{\tilde{g}}^d\right). \tag{24}$$

This means that every polynomial in $S_{g,\rho}^d$ can be represented by a sum of orthogonal coefficient tensors[2]

[1] It is possible to include rank adaptivity as in SALSA [21] or bASD [13] and we have noted this in the relevant places.

[2] The orthogonality comes from the symmetry of L which results in orthogonal eigenspaces.

$$\sum_{\tilde{g}=0}^{g} c^{(\tilde{g})} \quad \text{where} \quad Lc^{(\tilde{g})} = \tilde{g}c^{(\tilde{g})}. \tag{25}$$

There is however another, more compact, way to represent this function. Instead of storing $g + 1$ different tensor trains $c^{(0)}, \ldots, c^{(g)}$ of order d, we can merge them into a single tensor c of order $d + 1$ such that $c(i^d, \tilde{g}\ ) = c^{(\tilde{g})}(i^d)$. The summation over $\tilde{g}$ can then be represented by a contraction of a vector of 1's to the $(d + 1)$-th mode. To retain the block-sparse representation we can view the $(d + 1)$-th component as an artificial component representing a shadow variable x_{d+1}.

Remark 4.1. The introduction of the shadow variable x_{d+1} contradicts the locality assumptions of Theorem 3.4 and implies that the worst case rank bounds must increase. This can be problematic since the block size contributes quadratically to the number of parameters. However, a proof similar to that of Theorem 3.4 can be made in this setting and one can show that the bounds remain independent of d

$$\rho_{k,\tilde{g}} \leq \underline{1+} \sum_{\ell=1}^{K_{\text{loc}}} \min\left\{ \binom{K_{\text{loc}} - \ell + 1 + \tilde{g} - 2}{K_{\text{loc}} - \ell}, \binom{\ell \ \underline{+1}\ + g - \tilde{g} - 2}{\ell \ \underline{+1}\ - 1} \right\} \tag{26}$$

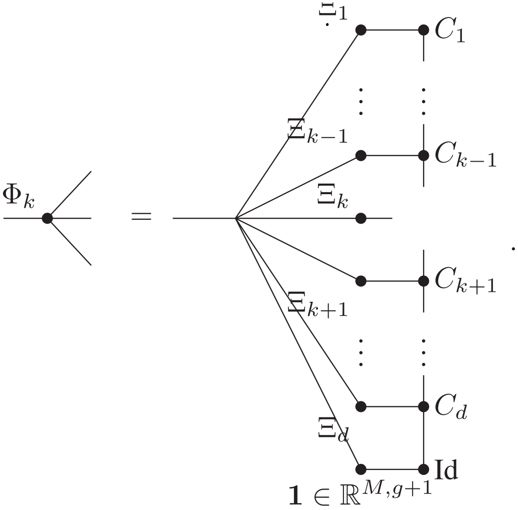
where the changes to **Eq. 22** are underlined. This is crucial, since in practice one can assume locality by physical understanding of the problem at hand. With this statement, we can guarantee that the ranks are only slightly changed by the auxiliary contraction and the locality property is not destroyed.

We denote the set of polynomials that results from this augmented block-sparse tensor train representation as

$$S_{g,\rho}^{d,\text{aug}} \tag{27}$$

where again ρ provides a bound for the block-size in the representation.

Since $S_{g,\rho}^{d,\text{aug}}$ is defined analogously to $B_\rho(W_g^d)$ we can use **Algorithm 1** to solve the related least-squares problem by changing the contraction **Eq. 16** to



To optimize the coefficient tensors $c^{(0)}, \ldots, c^{(g)}$ in the space $S_{g,\rho}^d$ we resort to an alternating scheme. Since the coefficient tensors are mutually orthogonal we propose to optimize each $c^{(\tilde{g})}$ individually while keeping the other summands $\{c^{(k)}\}_{k \neq \tilde{g}}$ fixed. This means that we solve the problem

$$u^{(\tilde{g})} = \underset{u \in W_{\tilde{g}}^d}{\operatorname{argmin}} \frac{1}{M} \sum_{m=1}^{M} \left\| f\left(x^{(m)}\right) - \sum_{\substack{k=0 \\ k \neq \tilde{g}}}^{g} u^{(k)}\left(x^{(m)}\right) - u\left(x^{(m)}\right) \right\|_{\text{F}}^2 \tag{29}$$

which can be solved using **Algorithm 1**. The original problem **Eq. 14** is then solved by alternating over g until a suitable convergence criterion is met. The complete algorithm is summarized in **Algorithm 2**.

The proposed representation has several advantages. The optimization with the tensor train structure is computationally less demanding than solving directly in S_g^d. Let $D = \dim(S_g^d) = \binom{d}{d+g}$. Then a reconstruction on S_g^d requires to solve a linear system of size $M \times D$ while a microstep in an ALS sweep only requires the solution of systems of size less than Mpr^2 (depending on the block sizes $\rho_{k,\tilde{g}}$). Moreover, the stack contractions as shown in *Least Squares* also benefit from the block sparse structure. This also means that the number of parameters of a full rank r tensor train can be much higher than the number of parameters of several $c^{(m)\prime}$s which individually have ranks that are even larger than r.

Remark 4.2. Let us comment on the practical pondering behind choosing $S_{g,\rho}^{d,\text{aug}}$ or $S_{g,\rho}^d$ by stating some pros and cons of [1]these parametrizations. We expect that solving the least-squares problem for $S_{g,\rho}^{d,\text{aug}}$ will be faster than for $S_{g,\rho}^d$ since it is computational more efficient to optimize all polynomials simultaneously than every degree individually in an alternating fashion. On the other hand, the hierarchical scheme of the summation approach may allow one to utilize multi-level Monte Carlo approaches. Together with the fact that every degree g possesses a different optimal sampling density this may result in a drastically improved best case sample efficiency for the direct method. Additionally, with $S_{g,\rho}^d$ it is easy to extend the ansatz space simply by increasing g which is not so straight-forward for $S_{g,\rho}^{d,\text{aug}}$. Which approach is superior depends on the problem at hand.

Algorithm 1 | Extended ALS (SALSA) for the least-squares problem on $B_\rho(W_g^d)$

```
input: Data pairs $(x^{(m)}, y^{(m)}) \in \mathbb{R}^d \times \mathbb{R}$ for $m = 1, \ldots, M$, a function dictionary $\Psi$,
       a maximal degree $g$, and a maximal block size $\rho$.
output: Coefficent tensor $c$ of a function $u \in B(W_g^d)$ that approximates the data.
For $k = 1, \ldots, d$ compute $\Xi_k$ according to Eq. 15;
Initialize the coefficient tensor $c$ for $u \in B(W_g^d)$;
Initialize SALSA parameters;
while not converged do
    Right orthogonalize $c$;
    for $k = 1, \ldots, d$ do
        Compute $\Phi_k$ according to Eq. 16;
        Compute the index set $\mathcal{I}$ of the non-zeros components in $C_k$ according to Eq. 20;
        Update $C_k$ by solving the SALSA-regularized version of $\Phi_k (j, i^3) \cdot C_k(i^3) = y(j)$ restricted to $i^3 \in \mathcal{I}$;
        Left orthogonalize $C_k$ and adapt the $k_{\text{th}}$ rank while respecting block size bounds $\rho$ and Eq. 21;
    end
    Update SALSA parameters;
end
return $c$
```

Algorithm 2 | Alternating extended ALS (SALSA) for the least-squares problem on $S^d_{g,\rho}$

```
input: Data pairs $(x^{(m)}, y^{(m)}) \in \mathbb{R}^d \times \mathbb{R}$ for $m = 1, \ldots, M$, a function dictionary $\Psi$, a
       maximal degree $g$, and a maximal block size $\rho$.
output: Coefficent tensors $c^{(0)}, \ldots, c^{(g)}$ of a function $u \in S^d_{g,\rho}$ that approximates the
        data.
Initialize the coefficient tensors $c^{(\tilde{g})}$ of $u^{(\tilde{g})} \in B_\rho(W^d_{\tilde{g}})$ for $\tilde{g} = 0, \ldots, g$;
while not converged do
    for $\tilde{g} = 0, \ldots, g$ do
        Compute $z^{(m)} := y^{(m)} - \sum_{k \neq \tilde{g}} u^{(k)}(x^{(m)})$ for $m = 1, \ldots, M$;
        Update $c^{(\tilde{g})}$ by using Algorithm 1 on the data pairs $(x^{(m)}, z^{(m)})$ for $m = 1, \ldots, M$;
    end
end
return $c^{(\tilde{g})}$ for $\tilde{g} = 0, \ldots, g$
```

5 NUMERICAL RESULTS

In this section we illustrate the numerical viability of the proposed framework on some simple but common problems. We estimate the relative errors on test sets with respect to the sought function f and are interested in the required number of samples leading to recovery. Our implementation is meant only as a proof of concept and does not lay any emphasis on efficiency. The rank is chosen a priori, the stopping criteria are naïvely implemented and rank adaptivity, as would be provided by SALSA, is missing all together.[3] For this reason we only compare the methods in terms of degrees of freedom and accuracy and not in terms of computing time. These are relevant quantities nonetheless, since the degrees of freedom are often the limiting factor in high dimensions and the computing time is directly related to the number of degrees of freedom.

In the following we always assume $p = g + 1$. We also restrict the group sizes to be bounded by the parameter $\rho_{\max}$. In our experiments we choose $\rho_{\max}$ without any particular strategy but ideally, $\rho_{\max}$ would be determined adaptively by the use of SALSA, which we did not do in this work. For every sample size the error plots show the distribution of the errors between the 0.15 and 0.85 quantile. The code for all experiments has been made publicly available at https://github.com/ptrunschke/block_sparse_tt.

5.1 Riccati Equation

In this section we consider the closed-loop linear quadratic optimal control problem

$$\begin{aligned} &\underset{u}{\text{minimize}} && \|y\|^2_{L^2([0,\infty]\times[-1,1])} + \lambda\|u\|^2_{L^2([0,\infty])} \\ &\text{subject to} && \partial_t y = \partial_x^2 y + u(t)\chi_{[-0.4,0.4]}, \\ &(t,x) \in [0,\infty]\times[-1,1] && \\ & && y(0,x) = y_0(x),\ x \in [-1,1] \\ & && \partial_x y(t,-1) = \partial_x y(t,1) = 0 \end{aligned}$$

After a spatial discretization of the heat equation with finite differences we obtain a d-dimensional system of the form

$$\underset{u}{\text{minimize}} \int_0^\infty \boldsymbol{y}(t)^\top Q\boldsymbol{y}(t) + \lambda u(t)^2\,\mathrm{d}t \quad \text{subject to} \quad \dot{\boldsymbol{y}}$$
$$= A\boldsymbol{y} + Bu \quad \text{and} \quad \boldsymbol{y}(0) = \boldsymbol{y}_0.$$

TABLE 1 | Degrees of freedom for the full space W^d_g of homogeneous polynomials of degree g =2, the TT variant $B_{\rho_{\max}}(W^d_g)$ with maximal block size $\rho_{\max}$ =4, the space $T_r(V^d_p)$ with TT rank bounded by r =6, and the full space V^d_p for completeness.

W^8_2	$B_4(W^8_2)$	$T_6(V^8_3)$	V^8_3
36	94	390	6561

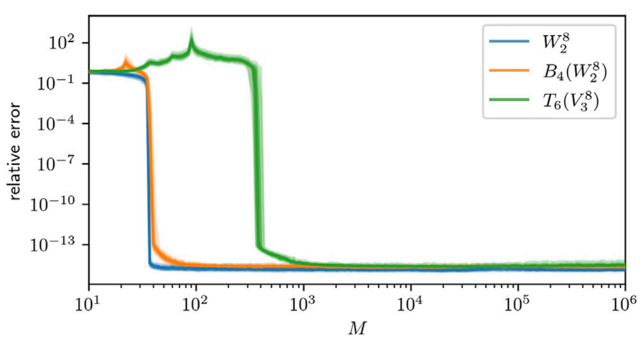


FIGURE 1 | 0.15–0.85 quantiles for the recovery error in W^8_2 (blue), $B_4(W^8_2)$ (orange), and $T_6(V^8_3)$ (green). The relative error is computed with respect to the L^2-norm using a Monte Carlo estimation with 10^6 samples.

It is well known [47] that the value function for this problem takes the form $v(\boldsymbol{y}_0) = \boldsymbol{y}_0^\top P\boldsymbol{y}_0$ where P can be computed by solving the *algebraic Riccati equation (ARE)*. It is therefore a homogeneous polynomial of degree 2. This function is a perfect example of a function that can be well-approximated in the space W^d_2. We approximate the value function on the domain $\Omega = [-1,1]^d$ for $d = 8$ with the parameters $g = 2$ and $\rho_{\max} = 4$.

In this experiment we use the dictionary of monomials $\Psi = \Psi_{\text{monomial}}$ (cf. **Eq. 4**) and compare the ansatz spaces W^8_2, $B_4(W^8_2)$, $T_6(V^8_3)$ and V^8_3. Since the function $v(x)$ is a general polynomial we use Lemma 3.3 to calculate the maximal block size 4. This guarantees perfect reconstruction since $B_4(W^8_2) = W^8_2$. The rank bound 6 is chosen s.t. $B_4(W^8_2) \subseteq T_6(V^8_3)$. The degrees of freedom of all used spaces are listed in **Table 1**. In **Figure 1** we compare the relative error of the respective ansatz spaces. It can be seen that the block sparse ansatz space recovers almost as well as the sparse approach. As expected, the dense TT format is less favorable with respect to the sample size.

A clever change of basis, given by the diagonalization of Q, can reduce the required block size from 4 to 1. This allows to extend the presented approach to higher dimensional problems. The advantage over the classical Riccati approach becomes clear when considering non-linear versions of the control problem that do not exhibit a Riccati solution. This is done in [8,9] using the dense TT-format $T_r(V^d_p)$.

5.2 Gaussian Density

As a second example we consider the reconstruction of an unnormalized Gaussian density

$$f(x) = \exp(-\|x\|^2_2).$$

TABLE 2 | Degrees of freedom for the full space S_g^d, the TT variant $S_{g,\rho_{\max}}^d$ with maximal block size $\rho_{\max}$ =1, the space $T_r(V_p^d)$ with TT rank bounded by r =1, the space $T_r(V_p^d)$ with TT rank bounded by r =8, and the full space V_p^d for completeness.

S_7^6	$S_{7,1}^6$	$T_1(V_8^6)$	$T_8(V_8^6)$	V_8^6
1716	552	48	2,176	262,144

again on the domain $\Omega = [-1,1]^d$ with $d = 6$. For the dictionary $\Psi = \Psi_{\mathrm{Legendre}}$ [cf. **Eq. 5**] we chose $g = 7$, $\rho_{\max} = 1$ and $r = 8$ and compare the reconstruction w.r.t. S_g^d, $S_{g,\rho_{\max}}^d$ and $T_r(V_p^d)$, defined in **(11)**, **(24)** and **(8)**. The degrees of freedom resulting from these different discretizations are compared in **Table 2**. This example is interesting because here the roles of the spaces are reversed. The function has product structure

$$f(x) = \exp\left(-x_1^2\right)\cdots\exp\left(-x_d^2\right)$$

and can therefore be well approximated as a rank 1 tensor train with each component C_k just being a best approximation for $\exp\left(-x_k^2\right)$ in the used function dictionary. Therefore, we expect the higher degree polynomials to be important. A comparison of the relative errors to the exact solution are depicted in **Figure 2**. This example demonstrates the limitations of the ansatz space S_7^6 which is not able to exploit the low-rank structure of the function f. Using $S_{7,1}^6$ can partially remedy this problem as can be seen by the improved sample efficiency. But since $S_{7,1}^6 \subseteq S_7^6$ the final approximation error of $S_{7,1}^6$ can not deceed that of S_7^6. One can see that the dense format $T_1(V_8^6)$ produces the best results but is quite unstable compared to the other ansatz classes. This instability is a result of the non-convexity of the set $T_r(V_p^d)$ and we observe that the chance of getting stuck in a local minimum increases when the rank r is reduced from 8 to 1. Finally, we want to address the peaks that are observable at $M \approx 500$ samples for $T_8(V_8^6)$ and $M \approx 1716$ samples for S_7^6. For this recall that the approximation in S_7^6 amounts to solving a linear system which is underdetermined for $M <$ 1716 samples and overdetermined for $M >$ 1716 samples. In the underdetermined case we compute the minimum norm solution and in the overdetermined case we compute the least-squares solution. It is well-known that the solution to such a reconstruction problem is particularly unstable in the area of this transition [39]. Although the set $S_{7,1}^6$ is non-linear we take the peak at $M \approx 500$ as evidence for a similar effect which is produced by the similar linear systems that are solved in the micro steps in the ALS.

5.3 Quantities of Interest

The next considered problem often arises when computing quantities of interest from random partial differential equations. We consider the stationary diffusion equation

$$\begin{aligned} \nabla_x a(x,y)\nabla_x u(x,y) &= f(x) \quad && x \in D \\ u(x,y) = 0 & && x \in \partial D \end{aligned}$$

on $D = [-1,1]^2$. This equation is parametric in $y \in [-1,1]^d$. The randomness is introduced by the uniformly distributed random variable $y \sim \mathcal{U}([-1,1]^d)$ that enters the diffusion coefficient

$$a(x,y) := 1 + \frac{6}{\pi^2}\sum_{k=1}^{d} k^{-2}\ \sin\left(\hat{\omega}_k x_1\right)\sin\left(\check{\omega}_k x_2\right)y_k$$

with $\hat{\omega}_k = \pi\left\lfloor\frac{k}{2}\right\rfloor$ and $\check{\omega}_k = \pi\left\lceil\frac{k}{2}\right\rceil$. The solution u often measures the concentration of some substance in the domain Ω and one is interested in the total amount of this substance in the entire domain

$$M(y) := \int_\Omega u(x,y)\,\mathrm{d}x.$$

An important result proven in [31] ensures the ℓ^p summability, for some $0 < p \leq 1$, of the polynomial coefficients of the solution of this equation in the dictionary of Chebyshev polynomials. This means that the function is very regular and we presume that it can be well approximated in S_g^d for the dictionary of Legendre polynomials Ψ_{Legendre}. For our numerical experiments we chose $d = 10$, $g = 5$ and $\rho_{\max} = 3$ and again compare the reconstruction w.r.t. S_g^d, the block-sparse TT representations of $S_{g,\rho_{\max}}^d$ and $S_{g,\rho_{\max}}^{d,\mathrm{aug}}$ and a dense

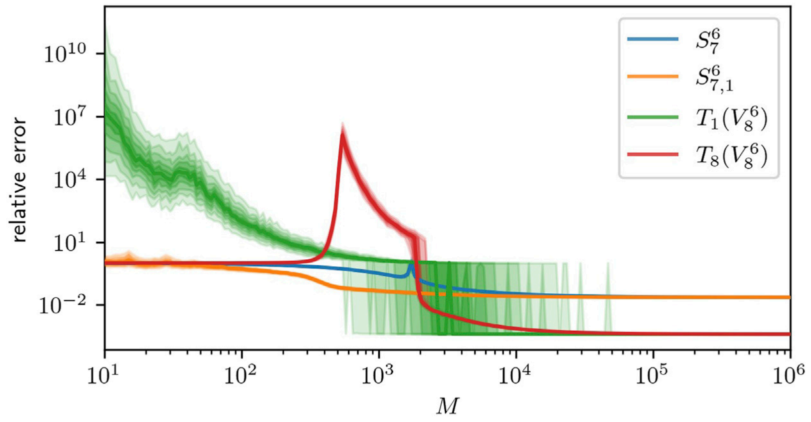


FIGURE 2 | 0.15–0.85 quantiles for the recovery error in S_7^6 (blue), $S_{7,1}^6$ (orange), $T_1(V_8^6)$ (green), and $T_8(V_8^6)$ (red). The relative error is computed with respect to the L^2-norm using a Monte Carlo estimation with 10^6 samples.

TABLE 3 | Degrees of freedom for the full space S_g^d, the TT variant $S_{g,\rho_{\max}}^d$ with maximal block size $\rho_{\max} = 3$, the space $T_r(V_p^d)$ with TT rank bounded by $r = 14$, and the full space V_p^d for completeness.

S_5^{10}	$S_{5,3}^{10}$	$S_{5,3}^{10,\mathrm{aug}}$	$T_{14}(V_6^{10})$	V_6^{10}
3,003	1726	803	7,896	60,466,176

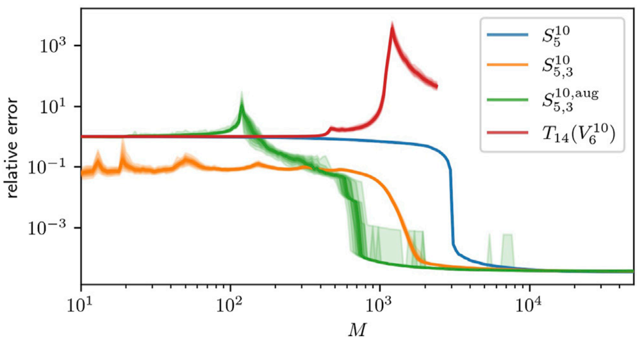


FIGURE 3 | 0.15–0.85 quantiles for the recovery error in S_5^{10} (blue), $S_{5,3}^{10}$ (orange), $S_{5,3}^{10,\mathrm{aug}}$ (green), and $T_{14}(V_6^{10})$ (red). The relative error is computed with respect to the L^2-norm using a Monte Carlo estimation with 10^6 samples. The experiment for $T_{14}(V_6^{10})$ was stopped early at $M = 1{,}200$ due to its prohibitive computational demand and because the expected behaviour is already observable.

TT representation of $T_r(V_p^d)$ with rank $r \leq 14$. Admittedly, the choice $d = 10$ is relatively small for this problem but was necessary since the computation on S_g^d took prohibitively long for larger values. A comparison of the degrees of freedom for the different ansatz spaces is given in **Table 3** the relative errors to the exact solution are depicted in **Figure 3**. In this plot we can recognize the general pattern that a lower number of parameters can be associated with an improved sample efficiency. However, we also observe that for small M the relative error for $S_{g,\rho}^d$ is smaller than for $S_{g,\rho}^{d,\mathrm{aug}}$. We interpret this as a consequence of the regularity of u since the alternating scheme for the optimization in $S_{g,\rho}^d$ favors lower degree polynomials by construction. In spite of this success, we have to point out that optimizing over $S_{g,\rho}^d$ took about 10 times longer than optimizing over $S_{g,\rho}^{d,\mathrm{aug}}$. Finally, we observe that the recovery in $T_{14}(V_6^{10})$ produces unexpectedly large relative errors when compared to previous results in [13]. This suggests that the rank-adaptive algorithm from [13] has a strong regularizing effect that improves the sample efficiency.

6 CONCLUSION

We introduce block sparsity [28,29] as an efficient tool to parametrize, multivariate polynomials of bounded degree. We discuss how to extend this to general multivariate polynomials of bounded degree and prove bounds for the block sizes for certain polynomials. As an application we discuss the problem of function identification from data for tensor train based ansatz spaces and give some insights into when these ansatz spaces can be used efficiently. For this we motivate the usage of low degree multivariate polynomials by approximation results (e.g. [30,31]) and recent results on sample complexity [32]. This leads to a novel algorithm for the problem at hand. We then demonstrate the applicability of this algorithm to different problems. Up until now block sparse tensor trains are not used for these recovery tasks. The numerical examples, however, demonstrate that at least dense tensor trains can not compete with our novel block-sparse approach. We observe that the sample complexity can be much more favorable for successful system identification with block sparse tensor trains than with dense tensor trains or purely sparse representations. We expect that the inclusion of rank-adaptivity using techniques from SALSA or bASD is straight forward, which we therefore consider an interesting direction from an applied point of view for forthcoming papers. We expect, that this would improve the numerical results even further. The introduction of rank-adaptivity would moreover alleviate the problem of having to choose a block size a-priori. Finally, we want to reiterate that the spaces of polynomials with bounded degree are predestined for the application of least-squares recovery with an optimal sampling density (cf [39]) which holds opportunities for further improvement of the sample efficiency. This leads us to the strong believe that the proposed algorithm can be applied successfully to other high dimensional problems in which the sought function exhibits sufficient regularity.

DATA AVAILABILITY STATEMENT

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found below: https://github.com/ptrunschke/block_sparse_tt.

AUTHOR CONTRIBUTIONS

MG is the main contributor of the block-sparse tensor train section. RS is the main contributor of the method section. PT is the main contributor of the sample complexity section, the implementation of the experiments and contributed to the proof of the main Theorem. All authors contributed to every section and contributed equally to the introduction and to the discussion.

FUNDING

MG was funded by DFG (SCHN530/15-1). RS was supported by the Einstein Foundation Berlin. PT acknowledges support by the Berlin International Graduate School in Model and Simulation based Research (BIMoS).

REFERENCES

1. Brunton SL, Proctor JL, and Kutz JN. Discovering Governing Equations from Data by Sparse Identification of Nonlinear Dynamical Systems. *Proc Natl Acad Sci USA* (2016) 113(15):3932–7. doi:10.1073/pnas.1517384113
2., Gelß P, Klus S, Eisert J, and Schütte C. Multidimensional Approximation of Nonlinear Dynamical Systems. *J Comput Nonlinear Dyn* (2019) 14(6). doi:10.1115/1.4043148
3. Goeßmann A, Götte M, Roth I, Ryan S, Kutyniok G, and Eisert J. *Tensor Network Approaches for Data-Driven Identification of Non-linear Dynamical Laws* (2020). NeurIPS2020 - Tensorworkshop.
4. Kazeev V, and Khoromskij BN. Low-Rank Explicit QTT Representation of the Laplace Operator and its Inverse. *SIAM J Matrix Anal Appl* (2012) 33(3): 742–58. doi:10.1137/100820479
5. Kazeev V, and Schwab C. Quantized Tensor-Structured Finite Elements for Second-Order Elliptic PDEs in Two Dimensions. *Numer Math* (2018) 138(1): 133–90. doi:10.1007/s00211-017-0899-1
6. Bachmayr M, and Kazeev V. Stability of Low-Rank Tensor Representations and Structured Multilevel Preconditioning for Elliptic PDEs. *Found Comput Math* (2020) 20(5):1175–236. doi:10.1007/s10208-020-09446-z
7. Eigel M, Pfeffer M, and Schneider R. Adaptive Stochastic Galerkin FEM with Hierarchical Tensor Representations. *Numer Math* (2016) 136(3):765–803. doi:10.1007/s00211-016-0850-x
8. Dolgov S, Kalise D, and Kunisch K. *Tensor Decomposition Methods for High-Dimensional Hamilton-Jacobi-Bellman Equations*. arXiv (2021). 1908.01533 [cs, math].
9. Oster M, Sallandt L, and Schneider R. *Approximating the Stationary Hamilton-Jacobi-Bellman Equation by Hierarchical Tensor Products*. arXiv (2021). arXiv: 1911.00279 [math].
10. Richter L, Sallandt L, and Nüsken N. *Solving High-Dimensional Parabolic PDEs Using the Tensor Train Format*. arXiv (2021). arXiv:2102.11830 [cs, math, stat].
11. Christian B, Martin E, Leon S, and Philipp T. *Pricing High-Dimensional Bermudan Options with Hierarchical Tensor Formats*. arXiv (2021). arXiv: 2103.01934 [cs, math, q-fin].
12. Glau K, Kressner D, and Statti F. Low-Rank Tensor Approximation for Chebyshev Interpolation in Parametric Option Pricing. *SIAM J Finan Math* (2020) 11(3):897–927. Publisher: Society for Industrial and Applied Mathematics doi:10.1137/19m1244172
13. Eigel M, Neumann J, Schneider R, and Wolf S. Non-intrusive Tensor Reconstruction for High-Dimensional Random PDEs. *Comput Methods Appl Math* (2019) 19(1):39–53. doi:10.1515/cmam-2018-0028
14. Eigel M, Schneider R, Trunschke P, and Wolf S. Variational Monte Carlo–Bridging Concepts of Machine Learning and High-Dimensional Partial Differential Equations. *Adv Comput Math* (2019) 45(5):2503–32. doi:10.1007/s10444-019-09723-8
15. Zhang Z, Yang X, Oseledets IV, Karniadakis GE, and Daniel L. Enabling High-Dimensional Hierarchical Uncertainty Quantification by Anova and Tensor-Train Decomposition. *IEEE Trans Comput.-Aided Des Integr Circuits Syst* (2015) 34(1):63–76. doi:10.1109/tcad.2014.2369505
16. Klus S, and Gelß P. Tensor-Based Algorithms for Image Classification. *Algorithms* (2019) 12(11):240. doi:10.3390/a12110240
17. Stoudenmire E, and Schwab DJ "Advances in Neural Information Processing Systems," in *Supervised Learning with Tensor Networks*. Editors D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett. Curran Associates, Inc. (2016) 29. Available at: https://proceedings.neurips.cc/paper/2016/file/5314b9674c86e3f9d1ba25ef9bb32895-Paper.pdf/
18. Oseledets IV. DMRG Approach to Fast Linear Algebra in the TT-Format. *Comput Methods Appl Math* (2011) 11(3):382–393. doi:10.2478/cmam-2011-0021
19. Holtz S, Rohwedder T, and Schneider R. The Alternating Linear Scheme for Tensor Optimization in the Tensor Train Format. *SIAM J Sci Comput* (2012) 34(2):A683–A713. doi:10.1137/100818893
20. White SR. Density Matrix Formulation for Quantum Renormalization Groups. *Phys Rev Lett* (1992) 69(19):2863–6. doi:10.1103/physrevlett.69.2863
21. Grasedyck L, and Krämer S. Stable ALS Approximation in the TT-Format for Rank-Adaptive Tensor Completion. *Numer Math* (2019) 143(4):855–904. doi:10.1007/s00211-019-01072-4
22. Holtz S, Rohwedder T, and Schneider R. On Manifolds of Tensors of Fixed TT-Rank. *Numer Math* (2012) 120(4):701–31. doi:10.1007/s00211-011-0419-7
23. Lubich C, Oseledets IV, and Vandereycken B. Time Integration of Tensor Trains. *SIAM J Numer Anal* (2015) 53(2):917–41. doi:10.1137/140976546
24. Chevreuil M, Lebrun R, Nouy A, and Rai P. A Least-Squares Method for Sparse Low Rank Approximation of Multivariate Functions. *Siam/asa J Uncertainty Quantification* (2015) 3(1):897–921. doi:10.1137/13091899x
25. Grelier E, Anthony N, and Chevreuil M. *Learning with Tree-Based Tensor Formats*. arXiv (2019). arXiv:1811.04455 [cs, math, stat].
26. Grelier E, Anthony N, and Lebrun R. *Learning High-Dimensional Probability Distributions Using Tree Tensor Networks*. arXiv (2021). arXiv:1912.07913 [cs, math, stat].
27. Haberstich C. *Adaptive Approximation of High-Dimensional Functions with Tree Tensor Networks for Uncertainty Quantification* (2020) Theses, École centrale de Nantes.
28. Singh S, Pfeifer RNC, and Vidal. G. Tensor Network Decompositions in the Presence of a Global Symmetry. *Phys Rev A* (2010) 82(5):050301. doi:10.1103/physreva.82.050301
29. Markus B, Michael G, and Max P. *Particle Number Conservation and Block Structures in Matrix Product States*. arXiv (2021). arXiv:2104.13483 [math.NA, quant-ph].
30. Breiten T, Kunisch K, and Pfeiffer L. Taylor Expansions of the Value Function Associated with a Bilinear Optimal Control Problem. *Ann de l'Institut Henri Poincaré C, Analyse non linéaire* (2019) 36(5):1361–99. doi:10.1016/j.anihpc.2019.01.001
31. Hansen M, and Schwab C. Analytic Regularity and Nonlinear Approximation of a Class of Parametric Semilinear Elliptic PDEs. *Mathematische Nachrichten* (2012) 286(8-9):832–60. doi:10.1002/mana.201100131
32. Eigel M, Schneider R, and Trunschke P. *Convergence Bounds for Empirical Nonlinear Least-Squares*. arXiv. arXiv:2001.00639 [cs, math], 2020.
33. Oliphant T. *Guide to NumPy* (2006).
34. Huber B, and Wolf S. *Xerus - A General Purpose* (2014). Tensor Library.
35. Espig M, Hackbusch W, Handschuh S, and Schneider R. Optimization Problems in Contracted Tensor Networks. *Comput Vis Sci.* (2011) 14(6): 271–85. doi:10.1007/s00791-012-0183-y
36. Oseledets IVV. Tensor-Train Decomposition. *SIAM J Sci Comput* (2011) 33(5):2295–317. doi:10.1137/090752286
37. Hackbusch W. *On the Representation of Symmetric and Antisymmetric Tensors. Preprint*. Leipzig, Germany: Max Planck Institute for Mathematics in the Sciences (2016).
38. Wolf S. *Low Rank Tensor Decompositions for High Dimensional Data Approximation, Recovery and Prediction*. [PhD thesis]. TU Berlin (2019).
39. Cohen A, and Migliorati G. Optimal Weighted Least-Squares Methods. *SMAI J Comput Math* (2017) 3:181–203. doi:10.5802/smai-jcm.24
40. Haberstich C, Anthony N, and Perrin G. *Boosted Optimal Weighted Least-Squares*. arXiv (2020). arXiv:1912.07075 [math.NA].
41. Göttlich S, and Schillinger T. *Microscopic and Macroscopic Traffic Flow Models Including Random Accidents* (2021).
42. Rasmussen CE, and Williams CKI. *Gaussian Processes for Machine Learning*. Cambridge: MIT Press (2006).
43. Cornford D, Nabney IT, and Williams CKI. Modelling Frontal Discontinuities in Wind fields. *J Nonparametric Stat* (2002) 14(1-2):43–58. doi:10.1080/10485250211392
44. Szalay S, Pfeffer M, Murg V, Barcza G, Verstraete F, Schneider R, et al. Tensor Product Methods and Entanglement Optimization Forab Initioquantum Chemistry. *Int J Quan Chem.* (2015) 115(19):1342–91. doi:10.1002/qua.24898
45. Michel B, and Nouy A. *Learning with Tree Tensor Networks: Complexity Estimates and Model Selection*. arXiv (2021). arXiv:2007.01165 [math.ST].
46. Ballani J, and Grasedyck L. Tree Adaptive Approximation in the Hierarchical Tensor Format. *SIAM J Sci Comput* (2014) 36(4):A1415–A1431. doi:10.1137/130926328
47. Curtain RF, and Hans Z. *An Introduction to Infinite-Dimensional Linear Systems Theory*. New York: Springer (1995).

48. Steinlechner M. Riemannian Optimization for High-Dimensional Tensor Completion. *SIAM J Sci Comput* (2016) 38(5):S461–S484. doi:10.1137/15m1010506

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.