Tensor network approaches for data-driven identification of non-linear dynamical laws

Alex Goeßmann
Institute of Mathematics
Technische Universität Berlin
10587 Berlin, Germany
goesmann@math.tu-berlin.de

Michael Götte
Institute of Mathematics
Technische Universität Berlin
10587 Berlin, Germany
goette@math.tu-berlin.de

Ingo Roth
Institute of Physics
Freie Universität Berlin
14195 Berlin, Germany
i.roth@fu-berlin.de

Ryan Sweke
Institute of Physics
Freie Universität Berlin
14195 Berlin, Germany
rsweke@gmail.com

Gitta Kutyniok
Mathematisches Institut
Ludwig-Maximilians-Universität München
80333 München, Germany
kutyniok@math.lmu.de

Jens Eisert
Institute of Physics
Freie Universität Berlin
14195 Berlin, Germany
jense@zedat.fu-berlin.de

Abstract

To date, scalable methods for data-driven identification of non-linear governing equations do not exploit or offer insight into the fundamental underlying physical structure. In this work, we show that various physical constraints can be captured via tensor network based parameterizations for the governing equation, which naturally ensures scalability. In addition to providing analytic results motivating the use of such models for realistic physical systems, we demonstrate that efficient rank-adaptive optimization algorithms can be used to learn optimal tensor network models without requiring a priori knowledge of the exact tensor ranks.

1 Introduction

Identifying a *governing equation* from observations of a physical system typically provides an explanatory understanding of the dynamics that goes significantly beyond the mere interpolation of observed data. For a traditional theorist, the process of inferring a suitable governing equation is guided by expert intuition and domain knowledge, which allows one to exploit physically motivated constraints. Automated approaches to this task have to balance the need for expressivity, which allows for the exploration of large theory spaces, with the need for scalability, which facilitates application to systems with many degrees of freedom. This is exactly the realm where the 'traditional approach' is bound to fail and automated approaches may turn out most valuable for consistent theory building.

To be more precise, let us model the state of a dynamical system by d real variables $(x_1, \dots, x_d) =: x$ and its time evolution by smooth trajectories $t \mapsto x(t) \in \mathbb{R}^d$. The dynamical laws of many physical systems are formulated by a suitable differential operator $\mathcal{D}$, such as the derivatives $\frac{d}{dt}$ or $\frac{d^2}{dt^2}$, acting

First Workshop on Quantum Tensor Networks in Machine Learning, 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

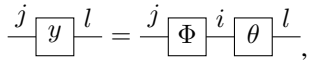
on the trajectory coordinatewise as

$$\mathcal{D}x(t) = \begin{bmatrix} \mathcal{D}x_1(t) \\ \mathcal{D}x_2(t) \\ \vdots \\ \mathcal{D}x_d(t) \end{bmatrix} =: \begin{bmatrix} f_1(x(t)) \\ f_2(x(t)) \\ \vdots \\ f_d(x(t)) \end{bmatrix} = f\big(x(t)\big)\,.$$

To learn the governing equation of a system, we assume access to potentially noisy estimates of different states x^j and the corresponding evaluation of the differential operator or, equivalently, $f(x^j)$. Such data can, for example, be obtained from time-series data by finite difference approximations of $\mathcal{D}$. Given a *hypothesis set* $\mathcal{F}$ of functions $\tilde{f} : \mathbb{R}^d \to \mathbb{R}$ on the state space, which represents the expert's intuition and the domain knowledge as a prior assumption, we formulate the recovery problem in the following fashion:

Problem 1 (Governing equation recovery)**.** *Identify the governing equation $f = [f_1 \dots f_d]$ with $f_k \in \mathcal{F}$ from the given observations $\{x^j, y^j := f(x^j)\}_{j=1}^m$.*

The interpolation of sample points is the central task of *supervised machine learning* (Bishop, 2006). Theory building, in contrast, is more ambitious and aims at the exact recovery of an *interpretable governing equation*, which motivates our approach. Simple hypothesis sets $\mathcal{F}$ are linear spaces spanned by *basis functions* $\{\phi_1(x), \dots, \phi_p(x)\}$. In this space, functions f_l are represented by their linear coefficients θ_{il} and function evaluations are modeled by multiplication with a dictionary matrix $\Phi_{ji} := \phi_i(x^j)$ as[1]



$$y_l^j = \sum_{i=1}^{p} \phi_i(x^j)\theta_{il}. \qquad (1)$$

Problem 1 then amounts to the linear inverse problem of recovering the coefficients θ_{il} representing the system $[f_1 \dots f_d]$. Since many dynamical systems need only a few basis functions to represent their governing equation, Ref. (Brunton et al., 2016) proposes the *sparse identification of non-linear dynamics (SINDy)*. A sparsity assumption on the coefficient vector effectively amounts to restricting $\mathcal{F}$ to the union of its lower-dimensional sub-spaces. In addition, it can resolve the ill-positioning of the linear inverse problem in the regime where Φ has a non-trivial kernel (Schaeffer et al., 2018). The assumption might also be interpreted as an implementation of the principle of *Occam's Razor*.

For systems with many degrees of freedom suitable linear function dictionaries for multivariate functions become inadmissibly large. This 'curse of dimensionality' yields a severe limitation for the scalability of SINDy to the regime where automated theory building would be most helpful. Similarly, approaches using *symbolic regression* (Schmidt & Lipson, 2009; Kusner et al., 2017; Ouyang et al., 2018; Li et al., 2019) for automated theory building will generate exponentially many combinations of basis functions, again severely limiting their applicability for systems with many degrees of freedom. This problem can be mitigated by *multi-linear* parameterization schemes (Cohen et al., 2015; Stoudenmire & Schwab, 2016; Levine et al., 2017; Gelß et al., 2019). At the heart of these parameterization schemes is the insight that multivariate function dictionaries often feature a tensor structure.

Although it is well known that one can break the curse of dimensionality associated with tensor representation by using tensor decompositions, such as tensor trains, in this work we seek to understand whether this is well motivated within the context of dynamical systems recovery. To this end, we show how one can reformulate dynamical systems recovery as a tensor recovery problem. Given this, we then determine how locality constraints on the interactions described by the governing equation, can be captured by different types of tensor network models. We substantiate our findings by studying classes of one-dimensional systems featuring local interactions, where we explicitly derive the ranks required for parameterization in the respective formats. Additionally, we show that the tensor recovery problem can be solved by state-of-the-art alternating least square (ALS) methods, such as SALSA. Importantly, the rank-adaptivity of these techniques ensures that assumptions on the locality of interactions are not a hard-coded restriction on the explored function space. As such, we

[1] We employ a common graphical notation, in which we sketch tensors of different orders by rectangles with legs representing its indices. Connections of legs to different tensor indicates scalar products performed in the respective index spaces, see App. A.

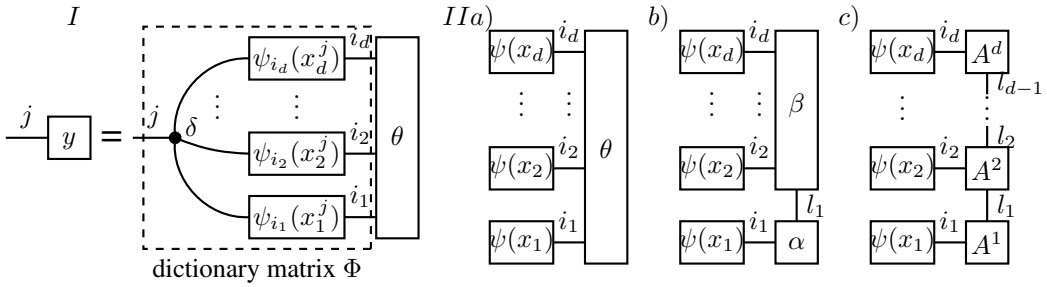


Figure 1: I - Tensor recovery as a linear inverse problem with observations y generated by contraction of the coefficient tensor θ on the dictionary matrix Φ (dashed). Employing a Hadamard product, represented by the contraction with a delta tensor δ, Φ is decomposed into univariate dictionary matrices $\psi_{i_k}(x_k^j)$. II - Parameterization of a function by (a) a generic tensor θ, which is in (b) decomposed into tensors α and β and in (c) decomposed into the tensor train format by iterative repetition of the decomposition.

do not require detailed a-priori knowledge about suitable ranks and tensor structures for a specific problem. This work therefore provides a general framework in which traditional concepts of theory building are combined with data-driven optimization algorithms through physical insights, facilitating progress towards efficient and broadly applicable automated theory discovery.

The remainder of this work is structured as follows: After introducing a tensor parameterization for multivariate functions in Section 2, we develop tensor network models for the coefficient vector in Section 3. Section 4 then introduces algorithms based on alternating least squares optimization and its rank-adaptive generalization, which will be tested in specific recovery tasks in Section 5.

2 Parameterization of multivariate functions and tensor recovery

To construct suitable hypothesis sets $\mathcal{F}$ for Problem 1 we assume a set of linearly independent univariate functions $\{\psi_i : \mathbb{R} \to \mathbb{R}\}_{i=1}^{\tilde{p}}$, from which we construct multivariate basis functions via

$$\phi_{i_1 \dots i_d}(x_1, \dots, x_d) := \psi_{i_1}(x_1) \cdot \psi_{i_2}(x_2) \cdot \ldots \cdot \psi_{i_d}(x_d)\,. \tag{2}$$

Given this, we then consider linear combinations of basis functions

$$f^\theta(x_1, \dots, x_d) := \sum_{i_1, \dots, i_d = 1}^{\tilde{p}} \theta_{i_1 \dots i_d} \phi_{i_1 \dots i_d}(x_1, \dots, x_d)\,, \tag{3}$$

which are fully specified by the coefficient tensor $\theta \in \mathbb{R}^{\tilde{p}^d}$. Note that the number $p = \tilde{p}^d$ of basis functions $\{\phi_{i_1 \dots i_d}\}$ constructed by this product ansatz grows exponentially in the number d of variables. This gives such function spaces a high expressivity but also manifests the *curse of dimensionality* that linear methods, such as SINDy, suffer for high-dimensional systems. In contrast to the sparsity hypothesis exploited in the SINDy method (Ho et al., 2018), we will in this work derive hypothesis sets that are invariant under span-preserving transformations of the basis functions. We will thus always be able to parameterize with respect to orthonormal basis functions, which ensures the equivalence of the scalar products on the coefficient space and the function spaces.

As illustrated in Fig. 1 I, the tensor product structure of the basis functions defined in Eq. (2) allows for calculation and storage of the dictionary matrix Φ (defined in Eq. (1)) with linear demand in d, via the Hadamard product of univariate dictionary matrices $\Psi^k := \psi_{i_k}(x_k^j)$. The evaluation of a function f^θ on the state x^j therefore amounts to a contraction of its coefficient tensors θ with the dictionary matrix Φ. Identifying for each function f_l in Problem 1 the corresponding coefficient tensor θ_l becomes therefore a multi-linear inverse problem. Our strategy to avoid the 'curse of dimensionality' is to restrict the exponentially large tensor space to a tractable hypothesis subset $\Theta \subset \mathbb{R}^p$, for which we pose the following problem:

Problem 2 (Tensor recovery)**.** *Given a dictionary matrix Φ and observations $y = \Phi\theta$, recover the tensor θ under the assumption $\theta \in \Theta$ with hypothesis set $\Theta \subset \mathbb{R}^p$.*

One approach to this problem proposed by Klus & Gelß (2019) understands the function (2) as a feature map lifting the states x into a tensor space $\mathbb{R}^p$. Then, inspired by the *representer theorem* (Hofmann et al., 2008) the subspace spanned by the represented states x^j can be taken as the hypothesis set Θ. The *MANDy method* originally introduced in Ref. (Gelß et al., 2019) efficiently identifies the minimal norm solution of the associated least squares problem by directly calculating the pseudo-inverse. This approach has the drawback that the hypothesis space itself is influenced by the data and does not directly take into account physical considerations. Our tensor network approach, which we will introduce in the next section, naturally overcomes this shortcoming by using locality as a guiding principle.

3 Tensor network hypothesis manifolds

Problem 2 naturally leads to the main question of this work: *What are natural hypothesis sets Θ?* In this section we develop tensor network models for the hypothesis set Θ that are informed by physical paradigms but remain highly adaptive.

3.1 Locality and separation ranks

In the context of tensor network representation of many-body quantum states, constraints on the structure of correlations relate directly to the locality of interactions in the physical system and provide a solid theoretical motivation for such representations (Schuch et al., 2008; Eisert et al., 2010). Motivated by these insights, we define here a notion of locality, founded on the decomposition properties of functions $f : \mathbb{R}^d \to \mathbb{R}$ into univariate functions, which represent the variables of a dynamical system.

The most extreme case is the absence of correlations: A multivariate function is called *separable*, if it is the product of univariate functions (Beylkin & Mohlenkamp, 2005). Functions f^θ parameterized by tensors θ (see Eq. (3)) are thus separable, if and only if the parameterizing tensor is elementary, i.e. it is an outer product $v^1 \otimes \cdots \otimes v^d$ of vectors $v^k \in \mathbb{R}^{\tilde{p}}$. While generic tensors do not satisfy such a decomposition, one can always decompose a tensor into a sum of elementary tensors, which is called a CP-decomposition. The CP-rank $r(\theta)$ of a tensor θ is the smallest number of elementary tensors appearing in a CP-decomposition. Since we have chosen linearly independent basis functions ψ_i, $r(\theta)$ determines the minimal number of separable functions for a decomposition

$$f^\theta(x_1, \ldots, x_d) = \sum_{l=1}^{r(\theta)} f^{v^{1,l}}(x_1) \cdot \cdots \cdot f^{v^{d,l}}(x_d), \tag{4}$$

where $v^{k,l} \in \mathbb{R}^{\tilde{p}}$ for $1 \leq k \leq d$ and $1 \leq l \leq r$. The CP-rank of a multivariate function provides a measure of correlations with respect to the partition $\{\{x_1\}, \ldots, \{x_d\}\}$ of its variables. Similarly, given an arbitrary partition $\mathcal{P}$ of the variables $\{x_1, \ldots, x_d\}$ into disjoint subsets, we define a separable function with respect to $\mathcal{P}$ as a function which can be written as the product of functions which depend only on variables in a subset of the partition. Accordingly, the separation rank $r_\mathcal{P}$ is then defined as the minimal number of separable functions, with respect to $\mathcal{P}$, such that f^θ can be decomposed into a sum of these functions. Guided by the conceptual framework of many-body physics (Eisert et al., 2010), we understand such separation ranks as a quantification of the correlation structure of a multivariate function. Note, however, that using $r_\mathcal{P}$ directly as a measure for correlation comes with stability issues related to the notion of border rank (Bini et al., 1980). While almost all tensors have full separation rank with respect to any partition, many functions arising in the sciences have collections of partitions with low separation ranks. This can be understood as arising from a notion of locality inherent in the underlying governing equations. In these cases, it is often possible to obtain a low-rank tensor network decomposition of the tensor θ encoding the function.

3.2 Tensor train format

A concrete example of an efficient tensor network parameterization with a clear understanding of the underlying locality structure is given by the *tensor train (TT)* format (see Fig. 1 II c), referred to as *matrix product state* in the many-body physics literature (Perez-Garcia et al., 2007). Given a collection of coordinates $\{x_1, \ldots, x_d\}$, the one-dimensional locality is captured in the partitions

$\mathcal{P}_k := \{\{x_1, \ldots, x_k\}, \{x_{k+1}, \ldots, x_d\}\}$, with low separation ranks $r_k := r_{\mathcal{P}_k}$. We find corresponding decompositions by reinterpreting the coefficient tensor θ as a matrix, via blocking the first k tensor indices into a column index and the last $d-k$ tensor indices into a row index (Holtz et al., 2012b), and then applying the singular value decomposition (SVD), we find r_k tensors α^{l_k} and β^{l_k} such that

$$f^\theta(x_1, \ldots, x_d) := \sum_{i_1,\ldots,i_d} \theta_{i_1 \ldots i_d} \psi_{i_1}(x_1) \ldots \psi_{i_d}(x_d) = \sum_{l_k=1}^{r_k} f^{\alpha_{l_k}}(x_1, \ldots, x_k) f^{\beta_{l_k}}(x_{k+1}, \ldots, x_d)\,.$$

Furthermore, performing the SVD iteratively, for increasing k, results in a representation of $\theta \in \mathbb{R}^p$ via *core tensors* $A^k \in \mathbb{R}^{r_{k-1} \times \tilde{p} \times r_k}$ (Oseledets, 2011). The collection of core tensors $\{A^k\}_{k=1}^d$, contracted along their common indices indicated in Fig. 1 IIc, is called a *tensor train (TT) representation* of θ. The minimum cardinalities r_k of the respective indices l_k for a TT representation of θ are equal to the separation ranks in Eq. (5). It has been further shown in (Holtz et al., 2012b), that the tensors θ, which are expressible in the tensor train format for given ranks $(r_1, \ldots, r_{d-1})$, constitute a sub-manifold Θ in the tensor space $\mathbb{R}^{\tilde{p}^d}$ with dimension

$$\dim \Theta = \sum_{k=1}^{d} r_{k-1} \tilde{p} r_k - \sum_{k=1}^{d-1} r_k^2 \ll \tilde{p}^d = \dim \mathbb{R}^{\tilde{p}^d}\,, \tag{5}$$

where we set $r_0 = r_d = 1$. This inequality holds, if d is large and the ranks r_k are bounded. Representing a tensor in the tensor train format with small ranks provides a way to store θ by its tensor train cores A^k. Assuming constant r_k, the memory demand scales linearly in d, which beats the 'curse of dimensionality' manifested in the exponential scaling of the tensor space dimension.

A particular dynamical system featuring one-dimensional locality is described by the *Fermi-Pasta-Ulam-Tsingou (FPUT)* equation, which models a vibrating string as a one-dimensional chain of masses. The variables of the system are the distortions of the masses, which are accelerated through nearest-neighbor interactions

$$\frac{d^2 x_l(t)}{dt^2} = f_l(x_1, \ldots, x_d) = (x_{l+1} - 2x_l + x_{l-1}) + \beta(x_{l+1} - x_l)^3 - \beta(x_l - x_{l-1})^3\,. \tag{6}$$

As we will show in App. C.1, each tensor θ_l, encoding the component f_l of the FPUT equation, can be exactly decomposed as a TT with ranks $r_{l-1} = r_l = 4$ and $r_k = 1$ for $|k - l| > 1$, which is a direct consequence of the locality manifested in the nearest-neighbor interactions (Gelß et al., 2019).

3.3 Correlations in the governing equation

The recovery of a multi-component governing equation $[f_1 \ldots f_d]$ can be formulated as d independent recovery problems, each of which aims to recover the TT representation of the respective coefficient tensor θ_l. Specifically, as illustrated in Fig. 2 Ia, we consider a collection of d different TT cores at each position, with the appropriate cores (and therefore TT) selected by a delta tensor δ. However, many governing equations show correlations *between* the functions f_l describing the dynamics of single variables, which is not taken into account by the "independent TT" model shown in Fig. 2 Ia. To illustrate this we recall the FPUT equation (6), in which each variable x_k is represented in the system $[f_1 \ldots f_d]$ in at most $n = 4$ different ways. We distinguish between the cases $k = l$, $k = l - 1$, $k = l + 1$ and $|k - l| > 1$, which we refer to as the *activation types* of the variable x_k. Assuming the knowledge of the activation pattern, that is the activation type $i_{kl} \in \{1, \ldots, n\}$ of the variable x_k in each function f_l, we introduce the model-specific selection tensor

$$S = \sum_{l=1}^{d} e^{(1)}_{i_{1,l}} \otimes \cdots \otimes e^{(d)}_{i_{d,l}} \otimes e_l \ \in \mathbb{R}^{n^d \times d}\,, \tag{7}$$

where we denote by $e_i^{(k)}$ the i'th basis vector in the k'th leg space of the tensor. The selection tensor S enables us to remove the redundancy resulting from small numbers of activation types in a governing equation. Each activation type corresponds with a specific tensor core, which is selected by contraction of S with the activation-informed cores A_k (Fig. 2 Ib). If the number n of activation types is small compared to the number d of functions, i.e., if $d > 4$ in the FPUT equation, this reparameterization results in an exponential decrease of the parameters.

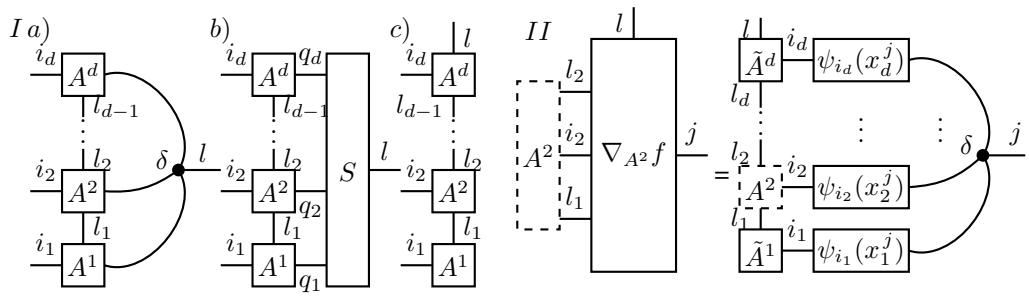


Figure 2: I - Tensor network models for a system of functions $[f_1 \dots f_d]$ with a one-dimensional locality. A common structure between the TT format of θ_l is exploited by the selection tensor S (b) with scaling advances compared to a naive delta tensor δ (a). If S is unknown, we add a single additional equation component index to the last TT core (c). II - Gradient of the function $f(\tilde{A}^1, \dots, \tilde{A}^d)$ with respect to a tensor train core A^2, computed by a tensor network contraction.

In many settings, one might assume small number of activation types without the precise knowledge of the selection tensor. In this situation one can parameterize the governing equation by a TT with an additional tensor leg representing the function index at one core tensor (Fig. 2 Ic). The TT-rank increase which is necessitated by such a model choice is quantified in the next section, for an exemplary class of systems.

3.4 One-dimensional interacting systems

We now demonstrate the above parameterization methods for classes of governing equations, which describe systems with a one-dimensional locality structure in their degrees of freedom.

Definition 1 (One-dimensional interacting systems)**.** *We say that a governing equation $[f_1 \dots f_d]$ is one-dimensional with* interaction range (s_1, s_2) *and* separation rank N*, if there exist a set of functions $\{g_i(x)\}_{i=1}^{\bar{p}}$ and for each $l \in \{1, \dots, d\}$ there is an index set $\mathcal{I}_l \subset \{1, \dots, \bar{p}\}^{s_1+s_2+1}$ with $|\mathcal{I}_l| \leq N$, such that*

$$f_l(x) = \sum_{(i_{l-s_1}, \dots, i_{l+s_2}) \in \mathcal{I}_l} g_{i_{l-s_1}}(x_{l-s_1}) \cdot \dots \cdot g_{i_{l+s_2}}(x_{l+s_2})$$

where we set $g_{i_k} = 1$ for $k \leq 0$ and $k \geq d+1$. We further call the univariate function spaces V spanned by the functions g_i and including the constant function 1 the leg embedding space *of f.*

For these classes of dynamical systems we derive the required ranks for the parameterization schemes, which have been discussed in Sec. 3.3.

Theorem 1 (Tensor representation of one-dimensional interacting systems)**.** *Let $[f_1 \dots f_d]$ be a one-dimensional governing equation with separation rank N and interaction range (s_1, s_2) and V its leg embedding space. Then, we have $f_l \in V^{\otimes d}$ for $l = 1, \dots, d$ and:*

(i) The CP*-rank of each f_l is at most N.*

(ii) Each f_l has a TT representation, Fig. 2 Ia, with ranks $r_k \leq N$ for $l - s_1 \leq k < l + s_2$ and $r_k = 1$ else.

(iii) There is a selection tensor S representation, Fig. 2 Ib, for $[f_1 \dots f_d]$ with $n = s_1 + s_2 + 2$ and TT ranks bounded by N.

(iv) The system $[f_1 \dots f_d]$ has a single TT representation, Fig. 2 Ic, with ranks $r_k \leq k - s_2 + 1 + N(s_1 + s_2)$.

We give the proof of Thm. 1 in App. B. The theorem shows that each f_l in a one-dimensional interacting system has bounded TT ranks independent of the number of variables d. Using a selection tensor S to represent $[f_1 \dots f_d]$ does not enlarge the TT ranks and allows for a total number of model parameters linear in d. Employing the single TT model instead of an explicit selection tensor, one finds a linear increase in the tensor train rank. As a consequence, the number of parameters scales as d^3 instead of d.

4 Optimization of tensor networks

We present in this section efficient learning algorithms for the proposed TT-type models. We begin by choosing an ℓ_2-norm loss-function and utilizing Eq. (1), which allows one to restate Problem 1 as a least-squares optimization problem in the parameterization θ:

$$\underset{\tilde{\theta}\in\mathbb{R}^{\tilde{p}\times\ldots\times\tilde{p}\times d}}{\text{minimize}} \|\Phi\tilde{\theta} - y\|_F^2 \quad \text{subject to} \quad \tilde{\theta} \in \Theta. \tag{P_Θ}$$

Here, we restricted the optimization to the function space encoded by a subset of tensors $\Theta \subset \mathbb{R}^{\tilde{p}\times\ldots\times\tilde{p}\times d}$, which is taken to be one of the tensor network models discussed in the previous section, and illustrated in Fig. 2 I. For all of the models Problem (P_Θ) is an optimization task over $\{\tilde{A}^k\}_k$, the TT cores that determine $\tilde{\theta}$. Hereinafter, we also use the symbol f for the function defined by $f(\tilde{A}^1, \ldots, \tilde{A}^d) \coloneqq \Phi\tilde{\theta}$, i.e. the value of the candidate function for f in the governing equation determined by the TT cores $\{\tilde{A}_k\}_k$ at the given observations x^j.

The *alternating least squares (ALS)* strategy (Holtz et al., 2012a) solves Problem (P_Θ) by alternatingly optimizing the single TT cores while regarding the other TT cores as constant. In each update step of the ALS algorithm, a tensor core $\tilde{A}^k$ is updated to a solution of

$$\underset{A^k\in\mathbb{R}^{r_{k-1}\times\tilde{p}\times r_k}}{\text{minimize}} \|f(\tilde{A}^1, \ldots, A^k, \ldots, \tilde{A}^d) - y\|_F^2. \tag{P_k}$$

After initialization by a random tensor in Θ, the algorithm iterates so-called *sweeps* in which the update step is performed for every tensor core, reminiscent of the density matrix renormalization group approach (White, 1992). The optimal update can be found by the minimum criterion of first order, i.e., as the solution of the linear equation

$$0 = \nabla_{A^k} \|f(\tilde{A}^1, \ldots, A^k, \ldots, \tilde{A}^d) - y\|_F^2 .$$

Due to the multi-linear dependence of the function f on the TT cores and the measurement tensor Φ, the analytical expression of the gradient $\nabla_{A^k} f$ is given by the tensor obtained via contraction of the measurement tensor Φ with all TT cores except for A^k (see Fig. 2 II). As explained in App. F each update step can thus efficiently be performed with computational complexity of $\mathcal{O}(r_{k-1}^3 r_k^3 \tilde{p}^3)$.

So far our formulation of the ALS algorithm relied on the separation ranks to be known. Physical principles such as locality might often merely justify the existence of a low-rank description, without providing precise upper bounds. In order for the framework presented here to be broadly applicable, it is thus necessary to utilize an optimization algorithm which is able to adaptively identify admissible low-ranks through the course of optimization. For the special case of the TT format, we demonstrate that the SALSA variant of the ALS algorithm by Grasedyck & Krämer (2019) is able to provide such rank-adaptivity.

5 Numerical simulations

Finally, we demonstrate the performance of ALS and its rank-adaptive generalization SALSA for the introduced tensor network parameterizations. To this end, we study variants of the Fermi–Pasta–Ulam–Tsingou (FPUT) equation, as introduced in Eq. (6), using a univariate dictionary consisting of the first $\tilde{p} = 4$ L_2-orthogonal Legendre polynomials.

We compare the recovery of the coefficient θ for two models: One involving the selection tensor S (Fig. 2 Ib) and the other regarding the single TT representation (Fig. 2 Ic). We use the regularized ALS algorithm to recover the model involving the selection tensor. The selection tensor has unit-rank tensors in its null space preventing rank-adaption in the sense of SALSA. Using the rank-adaptive SALSA, the single TT model is optimized without specifying the TT ranks.

We consider a coefficient tensor as successfully recovered if it relatively deviates from the correct coefficient tensor by less then 10^{-6} in Frobenius norm. The respective recovery rates for both models over ten experiments, for different values of d and m, are compared in Fig. 3. As expected, the required number of observations in both parameterization formats increases with the number of variables d. As per Thm. 1, note that while the selection tensor S allows for a model with constant TT ranks, the single TT model requires TT ranks that linearly increase with the position of the core tensors. Details concerning the exact ranks are provided in App. C.1. For small d, instances of

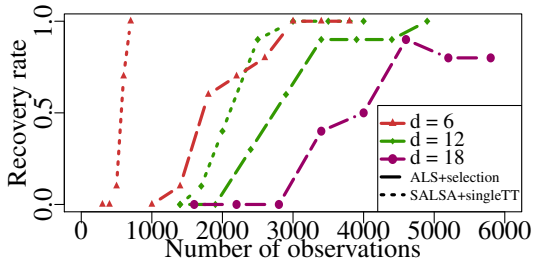


Figure 3: The recovery rate (relative error threshold 10^{-6}) for different number of observations m and dimension d for the FPUT equation with $\beta = 0.7$ using the selection tensor model (Fig. 2 Ib), trained with ALS for 15 iterations (long dash lines), and the single TT model (Fig. 2 Ic), trained with SALSA for 60 iterations (dotted lines). Each point is the average value of 10 trials.

the single TT model are already successfully recovered for smaller observation numbers compared to the selection tensor model, which reflects the more efficient parameterization with a single TT in this regime. The SALSA algorithm furthermore recovers the exact ranks of the model. For high d, however, increasing ranks of the single TT model cause an inadmissible computational demand, which scales for each update step quadratically in the dimension of the involved core tensor. Correspondingly, we observe in our numerics that performing the SALSA method on the single TT model becomes intractable on desktop hardware. In this regime, the model involving a selection tensor, with bounded ranks, is less computationally demanding. In conclusion, we find a trade-off between the computational cost and the amount of prior knowledge about the equations structure that the different models encode.

An extension of the FPUT equation that is still representable by the selection tensor model (Fig. 2 Ib) includes additional non-local mean field terms. ALS can be successfully employed for the recovery of this type of equations. We give detailed numerical results in the App. D. As the FPUT model is rather restrictive, we also studied the recovery of arbitrary one-dimensional interacting systems with interaction range $(1, 1)$ and random coefficients. As detailed in App. D, we still achieve adequate recovery rates with moderately sized numbers of observations for the selection tensor model (Fig. 2 Ib) with $d = 6$ and $d = 12$. For $d = 18$ the recovery becomes more involved due to local minima. In App. D we furthermore demonstrate the robustness of the recovery method under additive noise on the observations.

All simulations were implemented in python using `xerus` (Huber & Wolf, 2014–2020) and run on a computer with $4 \times 3500\,\mathrm{MHz}$ processors and 16 GB RAM. As an example for the selection tensor format a single simulation for $d = 18$ and $m = 6000$ takes about $500\,\mathrm{s}$ for 20 iterations.

6 Conclusion and outlook

In this work, we have provided a tensor network based framework for learning non-linear dynamical laws for systems with many degrees of freedom, in which the incorporation of fundamental physical principles, such as locality of admissible correlations, leads naturally to an efficient and scalable approach. In particular, we have shown that multivariate function dictionaries, built by products of univariate functions, give rise to tensor network structured representations for governing equations. Furthermore, we have argued that low separation ranks of these tensor networks are in accordance with constraints on the correlation structure of a system's variables, and as such have provided a generic scheme to exploit expert knowledge on the structure of governing equations. As a guiding example, we have discussed multiple tensor network models for systems with a constrained one-dimensional correlation structure, and obtained model-specific rank bounds for a large class of physical systems. We have furthermore adapted and implemented fixed-rank and rank-adaptive optimization schemes that do not require an a priori knowledge of the ranks. Our algorithms have numerically been demonstrated to be able to successfully recover variants of Fermi-Pasta-Ulam-Tsingou equation and random locally interacting systems. We regard our work as a fruitful step towards understanding the potential for tensor network parameterizations of governing equations in the light of fundamental physical principles.

While we here have exemplified a relation between one-dimensional locality and low-rank TT respresentation, we expect similar results to hold for higher-dimensional systems with projectively entangled pair states (Verstraete et al., 2008) and scale-invariant systems with the multi-scale renormalization ansatz (Vidal, 2007). Our approach furthermore admits the implementation of symmetry concepts, by constraining the structure of decomposing core tensors (Cohen et al., 2015; Schmoll et al., 2018). Additionally, the expressivity of our parameterization schemes can be improved by the selection of suitable dictionaries along the ideas of Champion et al. (2019) and Iten et al. (2020). Rigorous tensor recovery guarantees have been proven (Rauhut et al., 2015, 2016; Grotheer et al., 2019) and can be extended to the approach taken in this work. Finally, it is an interesting question whether one can also use physics-informed tensor network models to improve the scalability of alternative approaches to the recovery of dynamical systems (Williams et al., 2015; Kevrekidis et al., 2016; Kutz et al., 2016; Klus et al., 2018; Nüske et al., 2019). It is even conceivable to include tensor structured coefficients to increase the expressivity and scalability of symbolic regression algorithms.

Acknowledgements

We are grateful to Patrick Gelß, Stefan Klus and Christof Schütte for many fruitful discussions and extensive explanations about the SINDy and MANDy approaches. A. G. and G. K. acknowledge funding from the MATH+ project EF1-4. M. G. acknowledges funding from the DFG (SCHN 530/15-1). R. S. acknowledges funding from the Alexander von Humboldt foundation. I. R. and J. E. acknowledge funding from the DFG (EI 519/9-1 CoSIP, CRC 1114 project B06, MATH+ project EF1-7, CRC 183 project B01, and EI 519/15-1) and the BMWi (PlanQK).

References

Beylkin, G. and Mohlenkamp, M. J. Algorithms for numerical analysis in high dimensions. *SIAM Journal on Scientific Computing*, 26:2133–2159, 2005.

Bini, D., Lotti, G., and Romani, F. Approximate solutions for the bilinear form computational problem. *SIAM Journal on Computing*, 9(4):692–697, 1980.

Bishop, C. M. *Pattern recognition and machine learning*. Information science and statistics. Springer, New York, 2006.

Brunton, S. L., Proctor, J. L., and Kutz, J. N. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113:3932–3937, 2016.

Champion, K., Lusch, B., Kutz, J. N., and Brunton, S. L. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019.

Cohen, N., Sharir, O., and Shashua, A. On the expressive power of deep learning: A tensor analysis. *arXiv:1509.05009 [cs, stat]*, 2015.

Eisert, J., Cramer, M., and Plenio, M. B. Area laws for the entanglement entropy - a review. *Reviews of Modern Physics*, 82:277–306, 2010.

Fermi, E., Pasta, J., and Ulam, S. Studies of nonlinear problems. Technical Report Report No. LA-1940, Los Alamos Scientic Laboratory of the University of California, Los Alamos, NM, 1955.

Gelß, P., Klus, S., Eisert, J., and Schütte, C. Multidimensional approximation of non-linear dynamical systems. *Journal of Computational and Nonlinear Dynamics*, 14:061006–061006–12, 2019.

Glasser, I., Sweke, R., Pancotti, N., Eisert, J., and Cirac, J. I. Expressive power of tensor-network factorizations for probabilistic modeling, with applications from hidden markov models to quantum machine learning. In *Advances in Neural Information Processing Systems 32, Proceedings of the NeurIPS 2019 Conference*, 2019. arXiv:1907.03741.

Götte, M. `https://github.com/RoteKekse/systemrecovery`, 2020.

Grasedyck, L. and Krämer, S. Stable ALS approximation in the TT-format for rank-adaptive tensor completion. *Numerische Mathematik*, 143(4):855–904, 2019.

Grotheer, R., Li, S., Ma, A., Needell, D., and Qin, J. Iterative hard thresholding for low CP-rank tensor models. *arXiv:1908.08479 [cs, math, stat]*, 2019.

Ho, L. S. T., Schaeffer, H., Tran, G., and Ward, R. Recovery guarantees for polynomial approximation from dependent data with outliers. *arXiv:1811.10115 [cs, math, stat]*, 2018.

Hofmann, T., Schölkopf, B., and Smola, A. J. Kernel methods in machine learning. *The Annals of Statistics*, 36(3):1171–1220, 2008.

Holtz, S., Rohwedder, T., and Schneider, R. The alternating linear scheme for tensor optimization in the tensor train format. *SIAM Journal on Scientific Computing*, 34(2):A683–A713, 2012a.

Holtz, S., Rohwedder, T., and Schneider, R. On manifolds of tensors of fixed TT-rank. *Numerische Mathematik*, 120:701–731, 2012b.

Huber, B. and Wolf, S. Xerus - a general purpose tensor library. `https://libxerus.org/`, 2014–2020.

Iten, R., Metger, T., Wilming, H., del Rio, L., and Renner, R. Discovering physical concepts with neural networks. *Physical Review Letters*, 124:010508, 2020.

Kevrekidis, I., Rowley, C., and Williams, M. A kernel-based method for data-driven Koopman spectral analysis. *Journal of Computational Dynamics*, 2:247–265, 2016.

Klus, S. and Gelß, P. Tensor-based algorithms for image classification. *arXiv:1910.02150 [cs, stat]*, 2019.

Klus, S., Nüske, F., Koltai, P., Wu, H., Kevrekidis, I., Schütte, C., and Noé, F. Data-driven model reduction and transfer operator approximation. *Journal of Nonlinear Science*, 28:985–1010, 2018.

Kusner, M. J., Paige, B., and Hernández-Lobato, J. M. Grammar variational autoencoder. In *International Conference on Machine Learning*, pp. 1945–1954, 2017.

Kutz, J. N., Brunton, S. L., Brunton, B. W., and Proctor, J. L. *Dynamic mode decomposition*. Other Titles in Applied Mathematics. Society for Industrial and Applied Mathematics, 2016.

Levine, Y., Yakira, D., Cohen, N., and Shashua, A. Deep learning and quantum entanglement: Fundamental connections with implications to network design. *arXiv:1704.01552 [quant-ph]*, 2017.

Li, L., Fan, M., Singh, R., and Riley, P. Neural-guided symbolic regression with semantic prior. *arXiv:1901.07714 [cs, stat]*, 2019.

Nüske, F., Gelß, P., Klus, S., and Clementi, C. Tensor-based EDMD for the Koopman analysis of high-dimensional systems. *arXiv:1908.04741 [physics, stat]*, 2019.

Orús, R. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Annals of Physics*, 349:117–158, 2014.

Oseledets, I. V. Tensor-Train Decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

Ouyang, R., Curtarolo, S., Ahmetcik, E., Scheffler, M., and Ghiringhelli, L. M. SISSO: A compressed-sensing method for identifying the best low-dimensional descriptor in an immensity of offered candidates. *Physical Review Materials*, 2:083802, 2018.

Perez-Garcia, D., Verstraete, F., Wolf, M. M., and Cirac, J. I. Matrix product state representations. *Quantum Information and Computation*, 7:401, 2007.

Rauhut, H., Schneider, R., and Stojanac, Z. Tensor completion in hierarchical tensor representations. In Boche, H., Calderbank, R., Kutyniok, G., and Vybíral, J. (eds.), *Compressed Sensing and its Applications: MATHEON Workshop 2013*, Applied and Numerical Harmonic Analysis, pp. 419–450. Springer International Publishing, Cham, 2015.

Rauhut, H., Schneider, R., and Stojanac, Z. Low rank tensor recovery via iterative hard thresholding. *arXiv:1602.05217 [cs, math]*, 2016.

Schaeffer, H., Tran, G., and Ward, R. Extracting sparse high-dimensional dynamics from limited data. *SIAM Journal on Applied Mathematics*, 78:3279–3295, 2018.

Schmidt, M. and Lipson, H. Distilling free-form natural laws from experimental data. *Science*, 324: 81–85, 2009.

Schmoll, P., Singh, S., Rizzi, M., and Orus, R. A programming guide for tensor networks with global su(2) symmetry. 2018. arXiv:1809.08180.

Schuch, N., Wolf, M. M., Verstraete, F., and Cirac, J. I. Entropy scaling and simulability by matrix product states. *Physical Review Letters*, 100:030504, 2008.

Stoudenmire, E. and Schwab, D. J. Supervised learning with tensor networks. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29*, pp. 4799–4807. Curran Associates, Inc., 2016.

Verstraete, F., Cirac, J. I., and Murg, V. Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems. *Advances in Physics*, 57: 143, 2008.

Vidal, G. Entanglement renormalization. *Physical Review Letters*, 99:220405, 2007.

White, S. R. Density matrix formulation for quantum renormalization groups. *Physical Review Letters*, 69:2863, 1992.

Williams, M. O., Kevrekidis, I. G., and Rowley, C. W. A data–driven approximation of the Koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25:1307–1346, 2015.

Wolf, A. S. J. W. *Low rank tensor decompositions for high dimensional data approximation, recovery and prediction*. Doctoral thesis, Technische Universität Berlin, Berlin, 2019. doi: 10.14279/depositonce-8109.

Appendices

In these appendices we provide additional details concerning tensor network notation and the tensor representation of multivariate functions (App. A), as well as a proof of the expressivity result Theorem 1 (App. B). In App. C we further provide details on the types of governing equations that were studied in this work. Then, in App. D we provide additional numerical results, along with an extended discussion. Finally, in App. F we provide a detailed account of the algorithmic implementation of all numerical methods.

A Tensor notation

We briefly describe here the graphical notation used in this work to represent both tensors and tensor network contractions. This graphical notation is particularly common in the many-body physics literature (Verstraete et al., 2008; Eisert et al., 2010; Orús, 2014), and we refer there for a more detailed presentation. In general, an order d tensor $\theta \in \mathbb{R}^{\tilde{p}^d}$ is represented by a block with d different legs, each of which represents an index of the tensor,

$$\theta = \boxed{\theta}\ \begin{array}{|c|c|c|}\hline \ & \ & \cdots \\ \hline \end{array}\ . \tag{8}$$

The particular scalar elements $\theta_{i_1 \ldots i_d} \in \mathbb{R}$ of the tensor are then represented by indicating the appropriate indices at the relevant legs of the tensor, i.e.

$$\theta_{i_1 \ldots i_d} = \begin{array}{c}\boxed{\theta} \\ \begin{array}{|c|c|c|c|}\hline i_1 & i_2 & \cdots & i_d \\ \hline \end{array}\end{array}\ . \tag{9}$$

For example, a vector $\theta \in \mathbb{R}^p$ and a matrix $\Phi \in \mathbb{R}^{m\times p}$ would be represented by a box with one leg and two legs respectively,

$$\theta_i = \overset{i}{\boxed{\theta}} \qquad \Phi_{ji} = \overset{j}{\boxed{\Phi}}\overset{i}{} . \tag{10}$$

A contraction over a particular tensor index is then represented by connecting the appropriate legs of the relevant tensors. For example, matrix vector multiplication is indicated via

$$(\Phi\theta)_j = \sum_i \Phi_{ji}\theta_i = \overset{j}{\boxed{\Phi}}\overset{i}{}\boxed{\theta} . \tag{11}$$

Occasionally, when it is clear which index is being contracted, we will omit the summation index above the corresponding connected legs. This graphical notation therefore allows us to represent in a concise way tensors built from the contraction of component tensors, through networks of tensor diagrams connected in the appropriate fashion.

In this work we are primarily concerned with using tensor networks in order to represent multivariate functions, living in a linear space built from the tensor product of univariate function spaces. From a more general perspective consider a collection of linear spaces $\{V_k\}|_{k=1}^d$, corresponding with the univariate function spaces, each with a basis $\{e_i^{(k)}\}_{i=1}^p \subset V_k$. We can then define the tensor space $V = V_1 \otimes \cdots \otimes V_d$ by its basis, given by the tensor products of basis elements for the component linear spaces, i.e.

$$e_{i_1,\ldots,i_d} = e_{i_1}^{(1)} \otimes \cdots \otimes e_{i_d}^{(d)} . \tag{12}$$

With respect to this basis, an arbitrary element $v \in V$ of the tensor product space can then be expressed by the coefficient tensor $\theta \in \mathbb{R}^{p^d}$ which stores the coefficients of the components of v, i.e.

$$v = \sum_{i_1,\ldots,i_d} \theta_{i_1,\ldots,i_d} e_{i_1,\ldots,i_d} . \tag{13}$$

To build multivariate function spaces, we consider as building blocks vector spaces V_k of univariate functions, specified by basis functions $e_i^{(k)} \coloneqq \psi_i^{(k)} : \mathbb{R} \to \mathbb{R}, x \mapsto \psi_i^{(k)}(x)$. Multivariate functions are then constructed by tensor product of basis functions via

$$\psi_{i_1}^{(1)} \otimes \cdots \otimes \psi_{i_d}^{(d)} : (x_1, \ldots, x_d) \mapsto \psi_{i_1}^{(1)}(x_1) \cdot \cdots \cdot \psi_{i_d}^{(d)}(x_d) . \tag{14}$$

The tensor space $V = V_1 \otimes \cdots \otimes V_d$ is then defined by the linear hull of these tensor products. As a result, given d univariate function spaces $\{V_k\}$, we can specify an arbitrary element $f : \mathbb{R}^d \to \mathbb{R}$ of the tensor space $V = V_1 \otimes \cdots \otimes V_d$ via its coefficient tensor $\theta \in \mathbb{R}^{\tilde{p}^d}$. Explicitly, we have that

$$f(x_1, \ldots, x_d) = \sum_{i_1,\ldots,i_d} \theta_{i_1,\ldots,i_d} \psi_{i_1}^{(1)}(x_1) \cdot \psi_{i_2}^{(2)}(x_2) \ldots \psi_{i_d}^{(d)}(x_d). \tag{15}$$

Finally, note that we employ the symbol 1_d to represent the constant function with unit value, within the space of multivariate functions acting on d variables. It can be constructed by the tensor products of the constant functions $1 \in V_k$.

B Proof of Theorem 1

We provide here a proof of Theorem 1, which states rank bounds for the governing equation of a one-dimensional interacting system with range (s_1, s_2) and separation rank N. According to Def. 1, such a governing equation $[f_1 \ldots f_d]$ is of the form

$$f_l(x) = \sum_{(i_{l-s_1},\ldots,i_{l+s_2})\in\mathcal{I}_l} g_{i_{l-s_1}}(x_{l-s_1}) \cdot \ldots \cdot g_{i_{l+s_2}}(x_{l+s_2}). \tag{16}$$

Let V be the embedding space for f_l, i.e. an appropriate univariate function space from which to build the multivariate function space of which f_l is an element. Specifically, as explained in App. A, we

regard each function g_{i_k} as an element of the vector space V and thus $g_{i_1} \otimes \ldots \otimes g_{i_d}$ as a separable tensor in $V^{\otimes d}$. In order to be more precise we can make the dependence of f_l on all variables explicit by using the constant function (as described in App. A) for all variables outside of the interaction range. Eq. 16 then corresponds to

$$f_l = \sum_{(i_{l-s_1} \ldots i_{l+s_2}) \in \mathcal{I}_l} 1_{l-s_1-1} \otimes g_{i_{l-s_1}} \otimes \cdots \otimes g_{i_{l+s_2}} \otimes 1_{d-l-s_2} \,. \tag{17}$$

This is a CP-decomposition of the tensor f_l. Since we have $|\mathcal{I}_l| \leq N$, the CP-rank of f_l is therefore bounded by N, establishing the claim of (i).

For (ii) note that the minimal TT ranks for the representation of a tensor are equal to the separation ranks with respect to the partitions $\mathcal{P}_k = \{\{x_1, \ldots, x_k\}, \{x_{k+1}, \ldots, x_d\}\}$. A formal proof of this statement is given in Ref. (Holtz et al., 2012b). If $k < l - s_1$ the decomposition (17) is equal to

$$f_l = 1_k \otimes \left[\sum_{(i_{l-s_1}, \ldots, i_{l+s_2}) \in \mathcal{I}_l} 1_{l-s_1-k-1} \otimes g_{i_{l-s_1}} \otimes \cdots \otimes g_{i_{l+s_2}} \otimes 1_{d-l-s_2} \right] ,$$

thus the respective separation rank and, correspondingly, the TT rank r_k is at most one. This bound holds also for $k \geq l + s_2$, which follows from the analogous decomposition

$$f_l = \left[\sum_{(i_{l-s_1}, \ldots, i_{l+s_2}) \in \mathcal{I}_l} 1_{l-s_1-1} \otimes g_{i_{l-s_1}} \otimes \cdots \otimes g_{i_{l-s_2}} \otimes 1_{k-l-s_2} \right] \otimes 1_{d-k} \,.$$

For $k \in \{l - s_1, \ldots, l + s_2 - 1\}$ f_l does not permit such a factor decomposition in general but remains of the form

$$f_l = \sum_{(i_{l-s_1}, \ldots, i_{l+s_2}) \in \mathcal{I}_l} \left[1_{l-s_1-1} \otimes g_{i_{l-s_1}} \otimes \cdots \otimes g_{i_k} \right] \otimes \left[g_{i_{k+1}} \otimes \cdots \otimes g_{i_{l+s_2}} \otimes 1_{d-l-s_2} \right] .$$

We thus have a separation rank bounded as $r_k \leq |\mathcal{I}_l| \leq N$.

For (iii) we observe that at most only $s_1 + s_2 + 1$ different functions f_l depend non-trivially on the single variable x_k for all k. Thus, each variable x_k has $s_1 + s_2 + 1$ different activation types. The trivial dependence on a variable in all other functions constitutes another activation type. Hence, we need at most $n = s_1 + s_2 + 2$ activation types to represent the variables for the system $[f_1 \ldots f_d]$. We can thus build a selection tensor S by

$$S = \sum_{l=1}^{d} e_1^{(1)} \otimes \cdots \otimes e_1^{(l-s_1-1)} \otimes e_2^{(l-s_1)} \otimes \cdots \otimes e_{s_1+s_2+2}^{(l+s_2)} \otimes e_1^{(l+s_2+1)} \otimes \cdots \otimes e_1^{(d)} \otimes e_l \,.$$

Implementing an additional leg with dimension $n = s_1 + s_2 + 2$ indexing the different active TT cores A^k for each variable x_k results in a representation of $[f_1 \ldots f_d]$ by a contraction with S (Fig. 2 Ib). At each position k the required TT rank r_k is therefore bounded by the maximum separation rank of the single functions f_l, which is by (ii) bounded by N.

(iv) We represent the governing equation $[f_1 \ldots f_d]$ by the tensor $f \in V^{\otimes d} \otimes \mathbb{R}^d$,

$$f = \sum_{l=1}^{d} 1_{l-s_1-1} \otimes \tilde{f}_l \otimes 1_{d-l-s_2} \otimes e_l \,, \tag{18}$$

where we denote by $\tilde{f}_l$ the projection of f_l to the product space of its legs $l-s_1, \ldots, l+s_2$. Analogous to the proof of (ii), it is enough to bound the separation ranks r_k of the tensor f with respect to the partitions $\mathcal{P}_k$. For each $k = 1, \ldots, d$ we therefore split the sum (18) into the terms

$$f = \sum_{l \leq k-s_2} 1_{l-s_1-1} \otimes \tilde{f}_l \otimes 1_{d-l-s_2} \otimes e_l \tag{19}$$

$$+ 1_k \otimes \Big[\sum_{l > k+s_1} 1_{l-k-s_1-1} \otimes \tilde{f}_l \otimes 1_{d-l-s_2} \otimes e_l \Big] \tag{20}$$

$$+ \sum_{l=k-s_2+1}^{k+s_1} \sum_{(i_{l-s_1}, \ldots, i_{l+s_2}) \mathcal{I}_l} 1_{l-k-s_1-1} \otimes g_{i_{l-s_1}} \otimes \cdots \otimes g_{i_{l+s_2}} \otimes 1_{d-l-s_2} \otimes e_l \,. \tag{21}$$

The tensor in term (19) is nonzero if $k - s_2 > 0$ and has a separation rank with respect to $\mathcal{P}_k$ of at most $k - s_2$. Term (20) is nonzero in case $k < d - s_1$ and then contributes with a separation rank of at most 1. For each summand of the first sum of term (21) the corresponding tensor has a separation rank bounded by N, since $|\mathcal{I}_l| \leq N$. The number of these summands is given by $s_1 + s_2 - \max(s_1 - k + 1, 0) - \max(k - d - s_2, 0)$, thus bounded by $s_1 + s_2$. In all cases we have thus observed a separation rank bound of $r_k \leq k - s_2 + 1 + N(s_1 + s_2)$. □

C Examples of governing equations

We now discuss examples of governing equations, which illustrate the expressivity results and provide test systems for the numerical experiments. We follow closely the notation and techniques of Gelß et al. (2019).

C.1 Variants of the Fermi-Pasta-Ulam-Tsingou equation

Firstly we discuss variants of the *Fermi–Pasta–Ulam–Tsingou (FPUT)* equation, originally introduced by Fermi et al. (1955). Given constants m_l and β_l we define for $l = 1, \ldots, d$ the functions

$$\begin{aligned}\frac{d^2}{dt^2} x_l(t) = f_l(x(t)) =& \big(x_{l+1}(t) - 2x_l(t) + x_{l-1}(t)\big) + \beta_l \big(x_{l+1}(t) - x_l(t)\big)^3 \\ &- \beta_l \big(x_l(t) - x_{l-1}(t)\big)^3 + \sum_{\tilde{l}=1}^{d} m_{\tilde{l}} x_{\tilde{l}}\,.\end{aligned} \tag{22}$$

where $x_0 = x_{d+1} = 0$. Note that enforcing $\beta_l = \beta$ for all l results in a translationally invariant system of equations, and that this translation invariance can be broken by allowing for different values of β_l for each l. We will show in the following, that this does not affect the required ranks in the representation. Furthermore, in addition to the interaction terms, we have included the constant field term $\sum_{\tilde{l}=1}^{d} m_{\tilde{l}} x_{\tilde{l}}$.

We now derive an explicit parameterization of the functions f_l in the TT format (see Fig. 1 IIc). While in our numerical studies we represented the equations (22) with respect to $L_2([-1, 1])$-orthonormal basis functions (App. E), here we choose the monomials $\{1, x_k, x_k^2, x_k^3\}$ as basis functions $\{\psi_{i_k}\}$ for each variable x_k. As such, we remark that the representability of the coefficient tensor in the following tensor network formats is invariant under orthonormalization of the dictionary $\{\psi_i\}_{i=1}^{\tilde{p}}$ with respect to any scalar product.

The variable dependencies of Eq. (22) unravel the structure of the corresponding tensor θ_l to be

$$\theta_l = 1_{l-2} \otimes \tilde{\theta}_l \otimes 1_{d-l-1} + \sum_{\tilde{l}=1}^{d} 1_{\tilde{l}-1} \otimes m_{\tilde{l}} e_1^{(\tilde{l})} \otimes 1_{d-\tilde{l}}\,, \tag{23}$$

where $\tilde{\theta}_l$ is a tensor of order three and again we have denoted by 1 the constant function in the respective function spaces (see App. A). Factorization of the terms in Eq. (22) corresponds to a CP-decomposition of $\tilde{\theta}_l$ as

$$\begin{aligned}\tilde{\theta}_l =\ & e_1^{(l-1)} \otimes \Big[[-2e_2^{(l)} - 2\beta_l e_4^{(l)}] \otimes e_1^{(l+1)} + [e_1^{(l)} + 3\beta_l e_3^{(l)}] \otimes e_2^{(l+1)} - 3\beta_l e_2^{(l)} \otimes e_3^{(l+1)} + \beta_l e_1^{(l)} \otimes e_4^{(l+1)}\Big] \\ &+ e_2^{(l-1)} \otimes [e_1^{(l)} + 3\beta_l e_3^{(l)}] \otimes e_1^{(l+1)} \\ &- e_3^{(l-1)} \otimes 3\beta_l e_2^{(l)} \otimes e_1^{(l+1)} \\ &+ e_4^{(l-1)} \otimes \beta_l e_1^{(l)} \otimes e_1^{(l+1)}\,.\end{aligned}$$

To derive from the above CP-decomposition an illustration of the TT format of $\tilde{\theta}_l$ we build a matrix A^l by the coefficient vectors in the second leg space, which corresponds to the univariate function space of the variable x_l. Following Ref. (Gelß et al., 2019) the matrix is

$$A^l = \begin{bmatrix} -2e_2 - 2\beta_l e_4 & e_1 + \beta_l e_3 & -3\beta_l e_2 & \beta_l e_1 \\ e_1 + 3\beta_l e_3 & 0 & 0 & 0 \\ -3\beta_l e_2 & 0 & 0 & 0 \\ \beta_l e_1 & 0 & 0 & 0 \end{bmatrix}^{(l)}\,.$$

Defining further matrices $A^{l-1} = [e_1^{(l-1)}, e_2^{(l-1)}, e_3^{(l-1)}, e_4^{(l-1)}]$ and $A^{l+1} = [e_1^{(l+1)}, e_2^{(l-1)}, e_3^{(l-1)}, e_4^{(l+1)}]^T$, the tensor $\tilde{\theta}_l$ is given by the matrix contraction $A^{l-1} \cdot A^l \cdot A^{l+1}$, performed coordinate-wise as a tensor product (see Eq. (14)). We can now exploit the above tensor cores to build a TT decomposition of the full tensor θ_l. Specifically, we see that this requires TT ranks of $r_k = 4$ if $k = l-1, l$ and $r_k = 2$ for all other k:

$$\theta_l = \begin{bmatrix} e_1 & m_1 e_2 \end{bmatrix}^{(1)} \cdots \begin{bmatrix} e1 & m_{l-2}e_2 \\ 0 & e_1 \end{bmatrix}^{(l-2)} \cdot \begin{bmatrix} e_1 & e_2 & e_3 & e_4 \\ 0 & 0 & 0 & \beta_l^{-1} e_1 \end{bmatrix}^{(l-1)} \cdot$$

$$\cdot \begin{bmatrix} (-2+m_l)e_2 - 2\beta_l e_4 & (1+m_{k+1})e_1 + \beta_l e_3 & -3\beta_l e_2 & \beta_l e_1 \\ (1+m_{k-1})e_1 + 3\beta_l e_3 & 0 & 0 & 0 \\ -3\beta_l e_2 & 0 & 0 & 0 \\ \beta_l e_1 & 0 & 0 & 0 \end{bmatrix}^{(l)} \begin{bmatrix} 0 & e_1 \\ 0 & e_2 \\ 0 & e_3 \\ \beta_l^{-1} e_1 & e_4 \end{bmatrix}^{(l+1)} \cdot \tag{24}$$

$$\cdot \begin{bmatrix} e_1 & m_{l+2}e_2 \\ 0 & e_1 \end{bmatrix}^{(l+2)} \cdots \begin{bmatrix} m_d e_2 \\ e_1 \end{bmatrix}^{(d)} .$$

This explicit coefficient decomposition enables us to compute the ranks required to represent the FPUT equations via the different formats sketched in Fig. 2 I.

Remark 1 (Representation of the FPUT model with selection tensor S)**.** *Each variable x_k in the above decomposition of θ_l is represented by four different TT cores. The FPUT model in the variant* (22) *can thus be represented by a selection tensor S with $n = 4$ (see Fig. 2 Ib). The respective TT ranks are $r_k = 4$.*

Remark 2 (Representation of the FPUT model without selection tensor S)**.** *Representing the governing equation of the FPUT model by a single TT (see Fig. 2 Ic) amounts to the decomposition of the tensor*

$$\theta = \sum_{l=1}^{d} \theta_l \otimes e_l \in \mathbb{R}^{4 \times d \times d}.$$

If $m_l = 0$ and $\beta_l \neq 0$ the separation rank with respect to $\mathcal{P}_k = \{\{x_1, \ldots, x_k\}, \{x_{k+1}, \ldots, x_d, l\}\}$ is given by

$$r_k = \begin{cases} 4 & \text{if } k = 1 \\ 4 + k & \text{if } 1 < k < d \\ d & \text{if } k = d. \end{cases}$$

Although the variants of the FPUT equation with $m_l = 0$ are examples of a one-dimensional local interacting system with interaction range $(s_1, s_2) = (1, 1)$ and separation rank $N = 7$, we found representations with smaller ranks than estimated by Thm. 1. This is due to the special structure involving only nearest neighbor interactions (see Observation 3).

C.2 Polynomials with bounded degree

The structure of the FPUT equation that leads to a restricted TT rank of 4 is also present in a much larger class of functions, namely those that are quadratic forms of monomials of total degree smaller than $\tilde{p}$. This condition ensures that the terms involving the largest degree monomials are univariate. We summarize the statement for this class of functions as the following observation.

Remark 3 (A function class with constant TT-rank)**.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a function of the form*

$$f(x) = \sum_{l=1}^{d-1} \sum_{i+\tilde{i} < \tilde{p}} a(i, \tilde{i}, l) \cdot x_l^i x_{l+1}^{\tilde{i}}$$

where $a(i, \tilde{i}, l) \in \mathbb{R}$ are arbitrary coefficients. Further, let f be represented by $\theta \in \mathbb{R}^{\tilde{p} \times \cdots \times \tilde{p}}$ using the dictionary consisting of products of the monomials $x_l^i, i = 0, \ldots, \tilde{p} - 1$. Then, θ is of the form

$$\theta = \sum_{l=1}^{d-1} \sum_{i+\tilde{i} < \tilde{p}} a(i, \tilde{i}, l) \cdot 1_{l-1} \otimes e_i^{(l)} \otimes e_{\tilde{i}}^{(l+1)} \otimes 1_{d-l-1},$$

in particular, the TT ranks of θ are bounded by $\tilde{p}$.

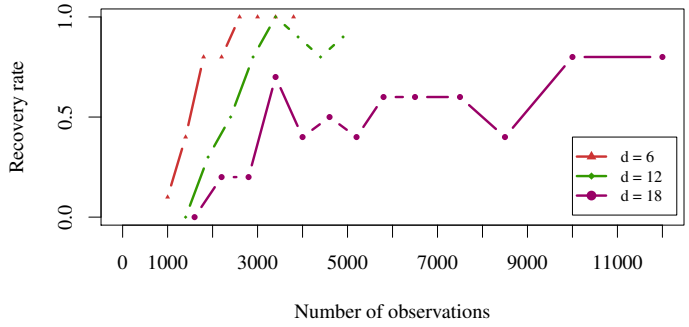


Figure 4: The recovery rate (relative error threshold 10^{-6}) for different number of observations m and dimension d for the FPUT equation with random β_l and random mean field parameters m_l (see C.1). Each point is the average value of 10 trials.

The proof of this statement is a straight-forward calculation of the function form, Fig. 1 II, using the given expression for θ. Similar results hold for any other univariate dictionary $\psi_i : \mathbb{R} \to \mathbb{R}, i = 0, \ldots, p-1$, which include the constant function 1.

C.3 Randomized local interaction model

In order to study generic models of one-dimensional local interacting systems, we focus here on randomly generated instances of systems with interaction range (s_1, s_2). Taking the first four Legendre polynomials as basis functions (see App. E), we take each equation f_l in a system $[f_1 \ldots f_d]$ to be of the structure

$$\tilde{f}_l(x_{l-1}, x_l, x_{l+1}) = \sum_{i_{l-1}, i_l, i_{l+1}=1}^{4} c_{i_{l-1} i_l i_{l+1}} \psi_{i_{l-1}}(x_{l-1}) \psi_{i_l}(x_l) \psi_{i_{l+1}}(x_{l+1}) \tag{25}$$

where $c_{i_{l-1} i_l i_{l+1}}$ are random coefficients, uniformly drawn on $[-1, 1]$ and we set $x_0 = x_{d+1} = 0$. By construction, and using the notation of App. C.1, we note that each equation can be represented by a tensor $\tilde{\theta}_l$ with TT decomposition

$$\tilde{\theta}_l = A^{l-1} \cdot A^l \cdot A^{l+1} ,$$

where we can choose core tensors with ranks 4 as

$$\begin{aligned} A^{l-1} &= \begin{bmatrix} e_1 & e_2 & e_3 & e_4 \end{bmatrix}^{(l-1)} , \\ A^l_{i_{l-1} i_{l+1}} &= \sum_{i_l=1}^{4} c_{i_{l-1} i_l i_{l+1}} e_{i_l}^{(l)} , \\ A^{l+1} &= \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix}^{(l+1)} . \end{aligned} \tag{26}$$

We further note that the number of nonzero coefficients $c_{i_{l-1} i_l i_{l+1}}$ is an upper bound for the CP-rank of $\tilde{\theta}_l$, and thus for the separation rank N of the generated system (see Def. 1).

D Extended numerical results

In this section we present and discuss further numerical results. While Fig. 3 shows the recovery rate for the FPUT equations (22) with fixed constant $\beta_l = 0.7$ and $m_l = 0$, we will in this section

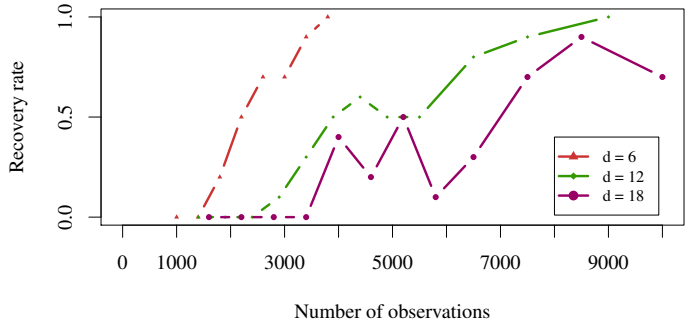


Figure 5: The recovery rate (relative error threshold 10^{-6}) for different numbers of observations m and dimensions d for random local interaction models trained with norm regularized ALS for 20 iterations (see C.3). Each point is the average of 10 trials.

turn our attention to more generic instances. To this end, we draw the coefficients β_l and m_l i.i.d. uniformly at random from the interval $[-1, 1]$. As explained in App. C.1 the equations can still be represented by a TT model with rank 4 using a selection tensor. We use this model trained by the norm-regularized ALS algorithm for the reconstruction (see also App.F.1). The resulting recovery rates averaged over 10 trials for different number of observations m are depicted in Fig. 4.

Going beyond the FPUT equation, the rank 4 TT model with selection tensor also allows us to describe local interacting models in the form of Eq. (25), and as such we proceed to explore this more general setting. To do so, we choose to use the first four Legendre polynomials (see App. E) for the local function set $\{\psi_i\}$. We then draw instances of these equations that have 20 non-vanishing coefficients c_{ijk} per equation, with support selected independently uniformly at random for each equation. The values of the non-trivial coefficients are drawn i.i.d. uniformly at random from the interval $[-1, 1]$. Using a sparse support allows us to keep the norm of $\tilde{f}$ moderately small. Fig. 5 shows the recovery rate for instances of such locally interacting equations for norm-regularized ALS. Each point is the average of 10 trial runs.

If a sufficient number of observations is provided, we observe a recovery rate close to 1 for both random equation types with small $d = 6, 12$ using ALS. The numerical results in the Fig. 3, 4 and 5 additionally demonstrate that for larger d, here $d = 18$, the recovery rates were poorly improving when increasing the number of observations m in the norm regularized ALS. In cases, where recovery was not achieved, especially for larger number of observations m, the convergence behavior of the relative error and the residual of the iterate θ_k indicated that ALS got stuck in local minima. This is a known issue with alternating optimization schemes like ALS. In the two randomized models, see App. C.1 and C.3, we have three random parts. Firstly, the governing equations have random coefficients. Secondly, the observations are drawn randomly, and thirdly the ALS like recovery schemes are initialized randomly. In order to test the intuition that successful recovery depends strongly on the random initialization we ran an additional simple experiment: If the error after 25 iterations of the regularized ALS was not less than 10^{-6}, we restarted the method, up to 5 times, with a different random initialization. For different numbers m of observations and $d = 18$ we get for the restarted version the recovery rates depicted in Table 1. We also state the averaged number of restarts, which is 4 in case of no success of the recovery method.

Finally, we demonstrate that the ALS algorithm is robust under noise. For that, we consider additive normal distributed noise on the observations of a FPUT model, i.e.

$$\frac{d^2 x_l(t)}{dt^2} = f_l(x_1, \ldots, x_d) = (x_{l+1} - 2x_l + x_{l-1}) + \beta(x_{l+1} - x_l)^3 - \beta(x_l - x_{l-1})^3 + \mathcal{N}(0, \sigma)\,. \tag{27}$$

where we will choose again $\beta = 0.7$ and $\mathcal{N}(0, \sigma)$ is the normal distribution with mean 0 and variance σ. For $d = 12$ and 4500 samples we run different recovery experiments for different signal

Dimension d	Number of observations m	Recovery Rate	Averaged Number of Restarts
18	1000	0 of 10	4
18	2000	1 of 10	3.8
18	3000	6 of 10	2.5
18	4000	10 of 10	1.3
18	5000	10 of 10	0.2
18	6000	10 of 10	0.3
18	7000	10 of 10	0.4

Table 1: Recovery rates for the restarted regularized ALS for randomized governing equations as in App. C.3. Again, recovery is achieved, if the relative error drops below 10^{-6}. The last column states the number of restarts (max. 5).

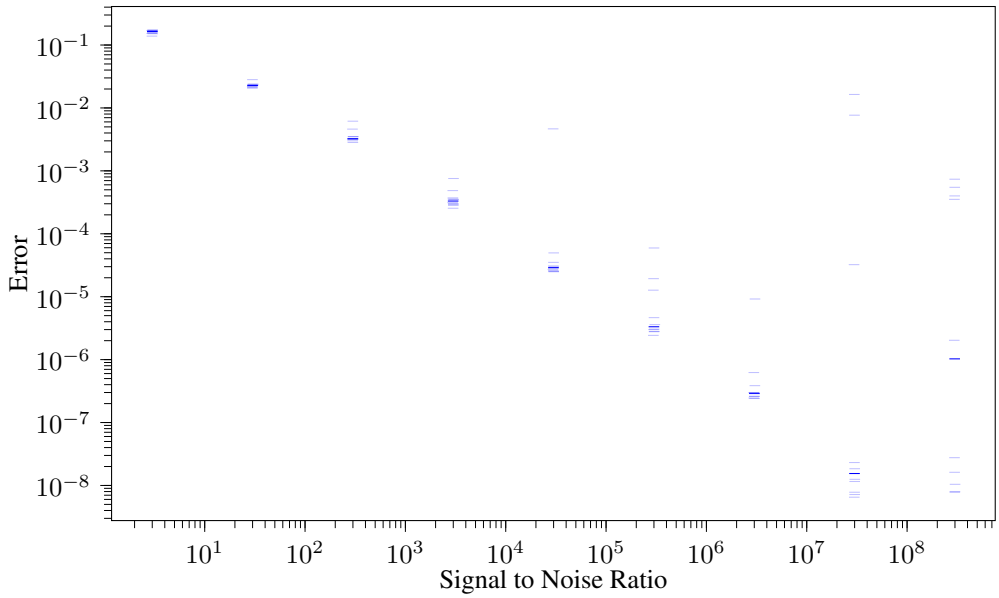


Figure 6: Relative error to the exact solution for the noisy FPUT model Equation (27) for dimension $d = 12$ and 4500 observations for different signal to noise ration. For exact data y and noisy data $y+\epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma)$ the signal to noise ratio is given by $\|y\|_F / \|\epsilon\|_F$. For each signal to noise ratio 10 runs were performed (brighter blue dashes). The darker blue dash depicts the median of these 10 runs.

to noise ratios. Please recall that for the noiseless case the recovery rate for 4500 samples was still not perfect (see Fig. 3). The results are depicted in Fig. 6. One sees that the noise level provides a lower bound to the possible accuracy and that in most runs such lower bounds are achieved.

E $L_2([-1, 1])$-orthogonal polynomials

As univariate basis functions we used in our simulations the Legendre polynomials

$$\begin{aligned}
\psi_1(x) &= 1, \\
\psi_2(x) &= x, \\
\psi_3(x) &= \frac{1}{2}(3x^2 - 1), \\
\psi_4(x) &= \frac{1}{2}(5x^3 - 3x).
\end{aligned}$$

The coefficients are chosen in such a way that the polynomials are L_2 orthogonal on $[-1, 1]$.

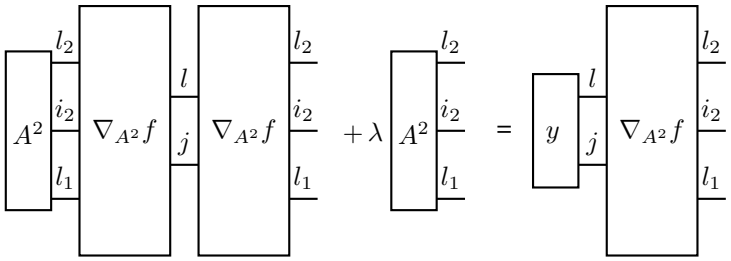


Figure 7: Least squares optimization of the tensor train component A^2, by the solution of a linear equation, the first order minimum condition of (P_k).

F Details on the numerical implementation

In this section we provide a detailed discussion of the numerical methods, ALS and SALSA, which were applied in our numerical experiments. We furthermore derive the complexity of the algorithms, which enables us to compare the computational requirements necessary for applying these algorithms with the different tensor models we have introduced (see Section 5). All code and numerical examples are openly available at the associated GitHub repository (Götte, 2020) (`https://github.com/RoteKekse/systemrecovery`).

F.1 Implementational notes on ALS

In the examples we have chosen to study, the selection tensor has a non-trivial low-rank kernel, and as such for the tensor network model involving the selection tensor problems can arise in the application of ALS due the potential for an increasing norm. In order to address this issue, we regularize the problem (P_k) by introducing a norm penalty in the following way

$$\min_{A^k} \|f(\tilde{A}^1, \dots, A^k, \dots, \tilde{A}^d) - y\|_F^2 + \lambda \|A^k\|_F^2 \,. \tag{P_k^{reg}}$$

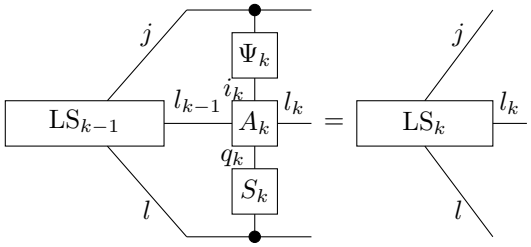
A core A^k optimizing the loss in (P_k^{reg}) is found by the first order condition (see Fig. 7):

$$\begin{aligned} 0 &= \nabla_{A^k} \|f(\tilde{A}^1, \dots, A^k, \dots, \tilde{A}^d) - y\|_F^2 + \lambda \nabla_{A^k} \|A^k\|_F^2 \\ &= 2\big[\langle A^k \nabla_{A^k} f, \nabla_{A^k} f \rangle - \langle y, \nabla_{A^k} f \rangle + \lambda A_k \big], \end{aligned} \tag{28}$$

which is just the solution of a linear equation system. We thereby used the scalar product corresponding to the Frobenius norm. In our numerical experiments, the results of which are seen in Fig. 3, 4 and 5, we initialized λ at $\lambda = 1$, and after each optimization sweep through all cores modified λ via the simple heuristic $\lambda_{\text{new}} = \lambda_{\text{old}}/10$. In the restarted version of the algorithm (described in App. D and Table 1) we used the following heuristic

$$\lambda_{\text{new}} = \min \left\{ 0.1 \frac{\|f(\tilde{A}^1, \dots, A^k, \dots, \tilde{A}^d) - y\|_F^2}{\|y\|_F \|A^k\|_F}, \frac{\lambda_{\text{old}}}{4} \right\}, \tag{29}$$

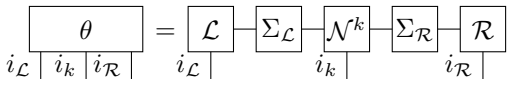
which aims at balancing the two terms in (P_k^{reg}).
For an efficient computation of the gradient $\nabla_{A^k} f$ (see Fig. 2 II), we use so called stacks (see Wolf (2019)). The benefit are achieved by a trade-off between memory and computational demand. We store the dictionary tensor as a list of d $m \times \tilde{p}$ matrices, therefore having a storage cost of $dm\tilde{p}$. For the examples in this work, the selection tensor can also be stored by a list of d $n \times d$ matrices with storage consumption $d^2 n$. The stacks used in the above numerical experiments, for one update step, then look as follows (here for the left stack):



One such stack update has complexity of $\mathcal{O}(md\tilde{p}r^2n)$ for the Format (2 Ib), where m is the number of observations, $\tilde{p}$ is the number of univariate basis functions, d is the number of variables and number of equations, r is the TT-rank of the given format and n is the number of activation patterns. If one can store the left and right stacks building the local linear equation systems (P_k) (see also Fig. 7) ALS schemes are much more efficient. Furthermore, the aforementioned local linear equation system's solution A_k of (28) has dimensions $r_{k-1}r_kn_k\tilde{p}$ for the selection tensor format (Fig. 2 Ib). Building the local linear operator has complexity $\mathcal{O}(md(\tilde{p}r^2n)^2)$. The complexity to solve the system of linear equations, using standard linear equation solvers, e.g. Gauss algorithm, is $\mathcal{O}((r^2n\tilde{p})^3)$. We therefore conclude that the contraction building the local linear operator is the most expensive part, if we can bound n and r.

F.2 Implementational notes on SALSA

The SALSA scheme, as introduced by Grasedyck & Krämer (2019), iteratively performs a rank-adaptive variant of the update step (P_{k-s}) (see Section 4). The major obstacle is that the solution of the ALS update step (P_k) is unstable under changes of the TT-rank. To resolve this instability problem, given a fixed k, Grasedyck & Krämer (2019) introduce a unique decomposition of a tensor θ into orthogonal tensors $\mathcal{L}, \mathcal{R}$ and diagonal matrices $\Sigma_{\mathcal{L}}, \Sigma_{\mathcal{R}}$,



,

such that $\mathcal{L}\Sigma_{\mathcal{L}}\big[\mathcal{N}\Sigma_{\mathcal{R}}\mathcal{R}\big]$ and $\big[\mathcal{L}\Sigma_{\mathcal{L}}\mathcal{N}\big]\Sigma_{\mathcal{R}}\mathcal{R}$ are singular value decompositions. To stabilize the update (P_k), one replaces the objective function by its average over a local neighborhood of θ with diameter ω. This is equivalent to the addition of regularization terms to the optimization problem:

$$\underset{\mathcal{N}^k \in \mathbb{R}^{r_{k-1}\times \tilde{p}\times r_k}}{\text{minimize}} \|f(\mathcal{L},\mathcal{N}^k,\mathcal{R}) - y\|_F^2 \quad + \omega^2 \left(\|\Sigma_{\mathcal{L},\epsilon}^{-1}\mathcal{N}^k\|_F^2 + \|\mathcal{N}^k\Sigma_{\mathcal{R},\epsilon}^{-1}\|_F^2 \right). \qquad (P_{k-\mathrm{s}})$$

For the numerical inversion of $\Sigma_{\mathcal{L}}$ and $\Sigma_{\mathcal{R}}$, the singular values below a certain threshold ϵ are set to ϵ. The rank-adaption strategy enforces a constant number $r_{\min}$ of singular values to be below ϵ for $\Sigma_{\mathcal{L}}$ and $\Sigma_{\mathcal{R}}$ in the following way: If after solving $(P_{k-\mathrm{s}})$ the number of singular values below ϵ increases or decreases, a corresponding number of randomly chosen singular vectors in $\mathcal{L}$ or in $\mathcal{R}$ are discarded or added, respectively.

The parameters ω and ϵ are adjusted during the iterations, where we state in the following our choices for the initalization and update rule. For the rank adaptive numerical experiment we used the initialization $\omega_{\mathrm{start}} = 1$ and $\epsilon_{\mathrm{start}} = 0.2$, furthermore we set $r_{\min} = 2$, $s_{\min} = 0.2$, $\omega_{\min} = 1.05$, $r_{\mathrm{start}} = (1, \ldots, 1)$ and $c = 0.01$. The parameters are updated in the following algorithm:

Stabilized alternating least squares approximation (SALSA)
Input: selection tensor S, dictionary tensor Φ, right hand side y
Output: iterate solution θ

1) Fix $r_{\min} \in \mathbb{N}$, $r = r_{\text{start}} \in \mathbb{N}^{d-1}$, $\epsilon = \epsilon_{\text{start}}, \omega = \omega_{\text{start}} > 0$.
2) Initialize randomly $\theta = A^1 \cdot \ldots \cdot A^d$, $A^k \in \mathbb{R}^{(1+r_{\min})\times\tilde{p}\times(1+r_{\min})}$
3) Solve ($P_{k-\text{s}}$) for all $k = 1, \ldots, d$ at least once for $A_k \in \mathbb{R}^{(r_{k-1}+r_{\min})\times\tilde{p}\times(r_k+r_{\min})}$.
4) For $k = 1, \ldots, d-1$ set the new rank $r_k^{(new)}$ to the number of singular values $\sigma_{k,j}, k = 1, \ldots, d-1, j = 1, \ldots, r_k + r_{\min}$ which are greater than ϵ.
5) If the rank r_k increased in step 4) add new singular values of size $c\epsilon, 0 < c < 1$ to the kth virtual index. This changes the tensor only little.
6) Decrease ω and ϵ. In our numerical experiments, we have used for $R(\theta) := \|\Phi\theta - y\|_F$

$$\omega_{\text{new}} = \min\{\sqrt{R(\theta)}, \frac{\omega_{\text{old}}}{\omega_{\min}}\},$$
$$\epsilon_{\text{new}} = s_{\min} R(\theta),$$

   which heuristically showed the best performance.
7) Repeat steps 3) to 6) until the residual $R(\theta)$ is smaller than some prescribed threshold.

Again, for an efficient implementation of the network contraction we can employ stacks. For the single TT format, similarly to the ALS case, the complexity of the stack updates is in $\mathcal{O}(m\tilde{p}dr^2)$ and building the local linear operator for the system of linear equations (see 7) is $\mathcal{O}(md(\tilde{p}r^2)^2)$. The computation of the solution of the linear equation system has a demand in $\mathcal{O}((r^2\tilde{p})^3)$.

We used SALSA for the single TT Format (Fig. 2 Ic) and by Theorem 1 the ranks increase linearly, while in the selection Format (Fig. 2 Ib) the ranks are bounded. It follows that for the example systems we study, the scaling of the update steps for the single TT formal is unfavourable for large d, which can be avoided by use of the selection tensor format.