KU LEUVEN

STADIUS
Center for Dynamical Systems, Signal Processing and Data Analytics

Citation/Reference	Vermeylen C., Van Barel M., **A Riemannian Rank-Adaptive Method for Higher-Order Tensor Completion in the Tensor-Train Format** Numerical Linear Algebra with Applications, vol. 32 (1), 2025, e2606.
Archived version	Author manuscript: the content is identical to the content of the published paper, but without the final typesetting by the publisher
Published version	https://doi.org/10.1002/nla.2606
Journal homepage	https://onlinelibrary.wiley.com/journal/10991506
Author contact	charlotte.vermeylen@.kuleuven.be
Abstract	A new Riemannian rank adaptive method (RRAM) is proposed for the low-rank tensor completion problem (LRTCP). This problem is formulated as a least-squares optimization problem on the algebraic variety of tensors of bounded tensor-train (TT) rank. The RRAM iteratively optimizes over fixed-rank smooth manifolds using a Riemannian conjugate gradient algorithm from Steinlechner. In between, the rank is increased by computing a descent direction selected in the tangent cone to the variety. A numerical method to estimate the rank increase is proposed. This numerical method is based on a new theoretical result for the low-rank tensor approximation problem and a definition of an estimated TT-rank. When the iterate comes close to a lower-rank set, the RRAM decreases the rank based on the TT-rounding algorithm from Oseledets and a definition of a numerical rank. It is shown that the TT-rounding algorithm can be considered an approximate projection onto the lower-rank set, which satisfies a certain angle condition to ensure that the image is sufficiently close to that of an exact projection. Several numerical experiments illustrate the use of the RRAM and its subroutines in Matlab. In all experiments, the proposed RRAM significantly outperforms the state-of-the-art RRAM for tensor completion in the TT format from Steinlechner in terms of computation time.
IR	https://lirias.kuleuven.be/4212166?limo=0

(article begins on next page)

Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE TYPE

A Riemannian rank-adaptive method for higher-order tensor completion in the tensor-train format

Charlotte Vermeylen[1] | **Marc Van Barel**[2]

[1]Department of Electrical Engineering (ESAT), KU Leuven, Belgium

[2]Department of Computer Science, KU Leuven, Belgium

Correspondence
Corresponding author Charlotte Vermeylen, Kasteelpark Arenberg 10, 3001 Leuven, Belgium.
Email: charlotte.vermeylen@kuleuven.be

Abstract

A new Riemannian rank adaptive method (RRAM) is proposed for the low-rank tensor completion problem (LRTCP). This problem is formulated as a least-squares optimization problem on the algebraic variety of tensors of bounded tensor-train (TT) rank. The RRAM iteratively optimizes over fixed-rank smooth manifolds using a Riemannian conjugate gradient algorithm from Steinlechner (2016) . In between, the rank is increased by computing a descent direction selected in the tangent cone to the variety. A numerical method to estimate the rank increase is proposed. This numerical method is based on a new theoretical result for the low-rank tensor approximation problem and a definition of an estimated TT-rank. When the iterate comes close to a lower-rank set, the RRAM decreases the rank based on the TT-rounding algorithm from Oseledets (2011) and a definition of a numerical rank. It is shown that the TT-rounding algorithm can be considered an approximate projection onto the lower-rank set which satisfies a certain angle condition to ensure that the image is sufficiently close to that of an exact projection. Several numerical experiments illustrate the use of the RRAM and its subroutines in Matlab. In all experiments, the proposed RRAM significantly outperforms the state-of-the-art RRAM for tensor completion in the TT format from Steinlechner (2016) in terms of computation time.

KEYWORDS
optimization, tensor-train, tensor completion, manifold, tangent cone

1 | INTRODUCTION

We consider the *low-rank tensor completion problem (LRTCP)* formulated as a least-squares optimization problem on the algebraic variety $\mathbb{R}^{n_1\times\cdots\times n_d}_{\leq(k_1,\ldots,k_{d-1})}$ of $n_1\times\cdots\times n_d$ real tensors of order d and TT-rank at most $k := (k_1,\ldots,k_{d-1})$ [18, Definition 1.4]:

$$\min_{X\in\mathbb{R}^{n_1\times\cdots\times n_d}_{\leq(k_1,\ldots,k_{d-1})}} \underbrace{\frac{1}{2}\left\|X_\Omega - A_\Omega\right\|^2}_{=:f_\Omega(X)}, \qquad A\in\mathbb{R}^{n_1\times\cdots\times n_d}, \tag{1}$$

where the sampling set $\Omega\subseteq\{1,\ldots,n_1\}\times\cdots\times\{1,\ldots,n_d\}$ and

$$Z_\Omega(i_1,\ldots,i_d) := \begin{cases} Z(i_1,\ldots,i_d) & \text{if } (i_1,\ldots,i_d)\in\Omega \\ 0 & \text{otherwise}\end{cases}, \qquad \forall\, Z\in\mathbb{R}^{n_1\times\cdots\times n_d}. \tag{2}$$

The norm in (1) is induced by the inner product [9, Example 4.149],

$$\langle Y, X\rangle = \langle \mathrm{vec}(Y), \mathrm{vec}(X)\rangle, \qquad \forall\, X, Y\in\mathbb{R}^{n_1\times\cdots\times n_d}, \tag{3}$$

Abbreviations: RRAM, Riemannian rank-adaptive method; TT, tensor-train; LRTCP, low-rank tensor completion problem; LRTAP, low-rank tensor approximation problem.

 © 2024 Copyright Holder Name

where $\text{vec}(Y) \in \mathbb{R}^{n_1 \cdots n_d}$ denotes the vectorization of a tensor. The elements of the vectorization are defined as:

$$\text{vec}(Y)(i_1 + (i_2-1)n_1 + \cdots + (i_d-1)n_1 \cdots n_{d-1}) := Y(i_1, \cdots, i_d), \qquad \forall\, i_j \in \{1, \cdots, n_j\},\ j = 1, \ldots, d. \tag{4}$$

The algebraic variety of tensor-train decompositions (TTDs) of bounded rank $k := (k_1, \ldots, k_{d-1})$ is defined as

$$\mathbb{R}^{n_1 \times \cdots \times n_d}_{\leq (k_1, \ldots, k_{d-1})} := \left\{ X \in \mathbb{R}^{n_1 \times \cdots \times n_d} \mid \text{rank}_{\text{TT}}\, X \leq (k_1, \ldots, k_{d-1}) \right\}, \tag{5}$$

where the inequality applies element-wise. A *tensor-train decomposition (TTD)* [22] of a tensor $X \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is a factorization $X = X_1 \cdot X_2 \cdots X_{d-1} \cdot X_d$, where $X_i \in \mathbb{R}^{r_{i-1} \times n_i \times r_i}$, and $r_0 := r_d := 1$. The '$\cdot$' indicates the multiplication of two tensors or a matrix and a tensor and more specifically the contraction between the last dimension of the first factor and the first dimension of the second factor: $X_i \cdot X_{i+1} := \left[X_i^{\text{R}} X_{i+1}^{\text{L}} \right]^{r_{i-1} \times n_i \times n_{i+1} \times r_{i+1}}$, where X_i^{R} and X_{i+1}^{L} are the right and left unfolding of a tensor respectively:

$$X_i^{\text{R}} := [X_i]^{r_{i-1} n_i \times r_i} := \text{reshape}\left(X_i, r_{i-1} n_i \times r_i\right), \qquad X_{i+1}^{\text{L}} := [X_{i+1}]^{r_i \times n_{i+1} r_{i+1}} := \text{reshape}\left(X_{i+1}, r_i \times n_{i+1} r_{i+1}\right).$$

Remark that matrices are invariant under left or right unfolding. An element in X can thus be obtained as

$$X(i_1, \ldots, i_d) = \sum_{j_1=1}^{r_1} \cdots \sum_{j_{d-1}=1}^{r_{d-1}} X_1(i_1, j_1) X_2(j_1, i_2, j_2) \cdots X_d(j_{d-1}, i_d).$$

The minimal $r := (r_1, \cdots, r_{d-1})$ for which a TTD of X exists, is called the *TT-rank* of X or $\text{rank}_{\text{TT}}\, X$. For second-order tensors (matrices), the TT-rank reduces to the standard matrix rank. An advantage of the TTD is that the rank can be determined as the matrix rank of the unfoldings:

$$r_i := \text{rank}\, X^{<i>}, \qquad X^{<i>} := [X]^{n_1 \cdots n_i \times n_{i+1} \cdots n_d}. \tag{6}$$

A TTD of minimal rank can be obtained by computing successive SVDs of the unfoldings [22, Algorithm 1]. The set of TTDs of fixed rank $(r_1, \ldots, r_{d-1})$ is known to be a *smooth manifold* [13, Lemma 4]:

$$\mathbb{R}^{n_1 \times \cdots \times n_d}_{(r_1, \ldots, r_{d-1})} := \left\{ X \in \mathbb{R}^{n_1 \times \cdots \times n_d} \mid \text{rank}_{\text{TT}}\, X = (r_1, \cdots, r_{d-1}) \right\}. \tag{7}$$

Different Riemannian optimization methods on the smooth manifold have already been developed [23, 4, 1]. However, they require as input an adequate value for the TT-rank which is difficult to determine for most applications [4, 15, 12], and is therefore in general determined by trial-and-error. Furthermore, in practical LRTCPs, A usually has full TT-rank due to noise. When k is set too high however, the complexity of an algorithm to solve (1) is unnecessarily high and overfitting can occur, i.e., X approximates A_Ω well but not the full tensor A. To detect overfitting, usually, a test data set Γ is used [7, 4, 32, 14, 26]. When the error on this test set increases during optimization while the error of (1) decreases overfitting has occurred and the algorithm should be stopped or the rank decreased. On the other hand, when k is set too low, the search space may not contain a sufficiently good approximation of A [26, 23, 14, 17]. It is thus important to choose an adequate value for k.

Compared to the *Tucker/higher-order singular value decomposition (HOSVD)* [30, 29, 6], the number of elements in a TTD only scales linearly in the order of the tensor d. Therefore, the TTD is especially useful for higher-order tensors. Higher-order tensor completion is used, e.g., for the approximation of high-dimensional functions and parameter-dependent PDEs [26, 25, 8]. More generally, higher-order tensor completion is useful in any application where a low-rank TT decomposition or approximation is desired and where it is costly or even impossible to obtain all elements. Low-rank TT approximations are also used for the efficient computation of other, optimization-based, decompositions such as the *canonical polyadic decomposition (CPD)* [5, 10, 35].

The sampling ratio of the tensor completion problem is defined as:

$$\rho_\Omega := \frac{|\Omega|}{n_1 \cdots n_d}, \tag{8}$$

where $|\Omega|$ denotes the number of elements in Ω. The smaller ρ_Ω, the more difficult it is to recover A from A_Ω by solving (1). When the samples are chosen randomly, the minimal number needed is not known [3, 2].

To resolve these difficulties, this paper proposes a new RRAM for higher-order tensor completion in the tensor-train format. RRAMs are state-of-the-art methods that can be used to minimize a continuously differentiable function on a low-rank variety,

a problem appearing, e.g., in low-rank matrix completion [36, 7]. The RRAMs in [36, 7] are developed for the set of bounded rank matrices. These RRAMs iteratively optimize over smooth fixed-rank manifolds starting from a low initial rank. They increase the rank by performing a line search along a descent direction selected in the *tangent cone* to the variety. This direction can be the projection of the negative gradient onto the tangent cone but does not need to; for instance, the RRAM developed by Gao and Absil [7] uses the projection of the negative gradient onto the part of the tangent cone that is normal to the tangent space. Additionally, they decrease the rank based on a truncated SVD when the fixed-rank algorithm converges to an element of a lower-rank set. This is possible because the manifold of fixed-rank matrices is not closed.

In this paper, we aim to generalize these RRAMs to the TT format. In this format, only the RRAM by Steinlechner is known to us from the literature [26]. This method is developed for high-dimensional tensor completion and has a random rank update mechanism in the sense that each TT-rank is increased subsequently by one by adding a small random term in the TTD of the current best approximation $X \in \mathbb{R}^{n_1 \times \cdots \times n_d}_{(r_1,\ldots,r_{d-1})}$:

$$X_i^{\mathrm{R}} \leftarrow \left[X_i^{\mathrm{R}} \;\; \varepsilon \, \mathtt{randn}\left(r_{i-1} n_i \times 1\right)\right], \qquad X_{i+1}^{\mathrm{L}} \leftarrow \begin{bmatrix} X_{i+1}^{\mathrm{L}} \\ \varepsilon \, \mathtt{randn}\left(1 \times n_{i+1} r_{i+1}\right) \end{bmatrix},$$

where $i \in \{1, \ldots, d-1\}$ and ε is small, e.g., 10^{-8}, such that f_Ω does not increase much, and `randn` is a built-in Matlab function to generate normally distributed pseudo-random numbers. The RRAM only terminates if a predefined maximal value for each r_i is reached. Furthermore, no rank reduction step is included, which makes the algorithm prone to overfitting. The full algorithm is available in the Manopt toolbox [1].

We improve this RRAM by including a method to increase the TT-rank based on a descent direction selected in the tangent cone. The tangent cone is the set of all tangent vectors to the variety and is discussed in more detail in Section 2.4. The full method to increase the rank is discussed in Section 3.1. Furthermore, Section 3.1.2 derives a numerical method to determine how much the rank should be increased based on a definition of an *estimated rank*. Lastly, a method to decrease the rank is given in Section 3.2, based on a definition of a *numerical rank*, which is necessary when the iterate comes close to a lower-rank set. This is possible because as for the manifold of fixed-rank matrices, the manifold of fixed-rank TTDs is not closed. This method can be considered as an *approximate projection* to the lower-rank set. The approximate projection ensures that the image is sufficiently close to that of the true projection. Approximate projections are discussed in more detail in Section 2.6. An overview of the proposed RRAM is given in Figure 1.

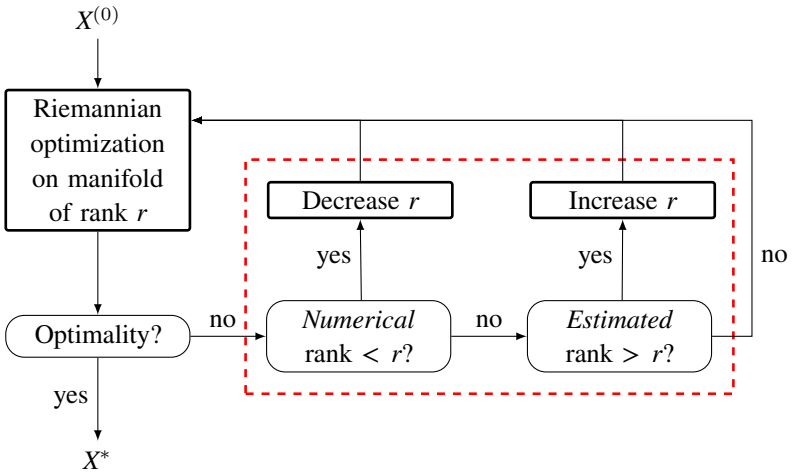

FIGURE 1 Illustration of the proposed RRAM.

Contrary to the random rank update mechanism proposed in [27], the method to increase the rank proposed Section 3.1 computes a descent direction and thus results in a decrease of the cost function. In [31], where a greedy rank update mechanism was proposed for the low-rank matrix completion problem, this strategy is called a *warm start* of the inner Riemannian algorithm. This strategy is compared with a random rank update and they observed the following: " ... *using a low-rank approximation of the gradient performs the best in terms of the number of iterations and total computational time. In addition, random vectors*

result in less predictable convergence compared to the gradient strategy." One of the experiments in Section 4.4 detangles the contribution of the projection and the rank estimation.

First, in Section 2 some preliminaries are given. Then, in Section 3.1 and Section 3.2, the methods to increase and decrease the rank are respectively proposed. Finally in Section 4, the full algorithm is given together with several numerical experiments to compare the proposed RRAM with the state-of-the-art RRAM [26].

2 | PRELIMINARIES

In this section, we first give the used notation and the preliminaries concerning the TTD. Afterward, we introduce a compact notation to simplify the tensor expressions derived in the rest of the paper. Fourthly, the parametrization of the tangent cone is discussed [18]. We derive slightly different orthogonality conditions such that no matrix inverse is needed in the method to increase the rank, which improves the overall stability of the RRAM. Then, in Section 2.5, the auxiliary *low-rank tensor approximation problem (LRTAP)* is defined. Lastly, approximate projections are discussed.

2.1 | Notation

Tensors are denoted by capital letters. A matrix can be considered as a tensor of order two. Matrices are therefore also denoted by capital letters. On the other hand, scalars and vectors, which are zero- and one-dimensional 'tensors' respectively, are denoted by lower-case letters. The order will always be clear from the context.

2.2 | Properties tensor-train decomposition

In this section, we review some properties of the TTD that are frequently used in the rest of the paper; we refer to the original paper [22] and the subsequent works [16, 26, 28] for more details. A third-order TTD is illustrated in Figure 2.

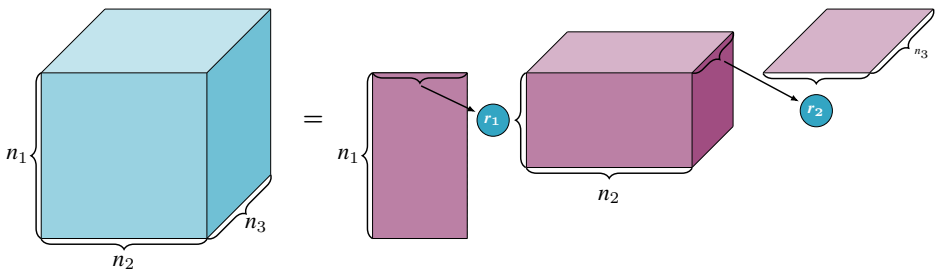

FIGURE 2 Illustration of a third-order TTD of size $n_1 \times n_2 \times n_3$ and TT-rank (r_1, r_2).

- A TTD is not unique:

$$X = (X_1 B_1) \cdot \left(B_1^{-1} \cdot X_2 \cdot B_2\right) \cdots \left(B_{d-2}^{-1} \cdot X_{d-1} \cdot B_{d-1}\right) \cdot \left(B_{d-1}^{-1} X_d\right),$$

where $B_i \in \mathrm{GL}(r_i)$, for $i = 1, \ldots, d-1$. Consequently, it can be proven that the dimension of the manifold in (7) is [13]

$$\dim\left(\mathbb{R}^{n_1 \times \cdots \times n_d}_{(r_1,\ldots,r_{d-1})}\right) = \sum_{i=1}^{d} r_{i-1} n_i r_i - \sum_{i=1}^{d-1} r_i^2. \tag{9}$$

Thus, similarly as for the SVD, *orthogonality conditions* can be enforced to improve the numerical stability of algorithms working with TTDs [22, 28, 26].

- To define orthogonality conditions, we let $\mathrm{St}(p,n) := \{U \in \mathbb{R}^{n\times p} \mid U^\top U = I_p\}$ denote the *Stiefel manifold*, where $n,p \in \mathbb{N}$ with $n \geq p$.
- For every $U \in \mathrm{St}(p,n)$, we let $P_U := UU^\top$ and $P_U^\perp := I_n - P_U$ denote the orthogonal projections onto the range of U and its orthogonal complement respectively.
- A TTD is called *i-orthogonal*, for $i \in \{1,\dots,d\}$, if

$$X = X_1' \cdots X_{i-1}' \cdot \dot{X}_i \cdot X_{i+1}'' \cdots X_d'' \tag{10}$$

where $X_j'^{\mathrm{R}} \in \mathrm{St}(r_j, r_{j-1}n_j)$, for $j = 1,\dots,i-1$, and $\left(X_k''^{\mathrm{L}}\right)^\top \in \mathrm{St}(r_{k-1}, n_k r_k)$, for $k = i+1,\dots,d$, and $\dot{X}_i \in \mathbb{R}^{r_{i-1}\times n_i \times r_i}$. The tensors X_i'' are called *left-orthogonal* and the tensors X_i' *right-orthogonal*. It holds that

$$\dot{X}_i^{\mathrm{R}} = X_i'^{\mathrm{R}} Q_i, \qquad \dot{X}_i^{\mathrm{L}} = R_i X_i''^{\mathrm{L}}, \tag{11}$$

for some $Q_i \in \mathrm{GL}(r_i)$ and $R_i \in \mathrm{GL}(r_{i-1})$. Thus,

$$Q_i = X_i'^{\mathrm{R},\top} \dot{X}_i^{\mathrm{R}}, \qquad R_i = \dot{X}_i^{\mathrm{L}} X_i''^{\mathrm{L},\top}, \tag{12}$$

where $X^{\mathrm{L},\top} := \left(X^{\mathrm{L}}\right)^\top$ and $X^{\mathrm{R},\top}$, and X can be written as

$$X = \dot{X}_1 R_2^{-1} \cdot \dot{X}_2 \cdot R_3^{-1} \cdot \dot{X}_3 \cdots R_d^{-1} \dot{X}_d = \dot{X}_1 Q_1^{-1} \cdot \dot{X}_2 \cdot Q_2^{-1} \cdot \dot{X}_3 \cdots Q_{d-1}^{-1} \dot{X}_d, \tag{13}$$

and consequently $Q_i = R_{i+1}$, for $i = 1,\dots,d-1$. We can derive a similar expression for $A \in \mathbb{R}^{n_1\times\cdots\times n_d}_{(r_1',\dots,r_{d-1}')}$:

$$A = \dot{A}_1 {Q_1'}^{-1} \cdot \dot{A}_2 \cdot {Q_2'}^{-1} \cdot \dot{A}_3 \cdots {Q_{d-1}'}^{-1} \dot{A}_d, \tag{14}$$

where $Q_i' = A_i'^{\mathrm{R},\top} \dot{A}_i^{\mathrm{R}} = \dot{A}_{i+1}^{\mathrm{L}} {A_{i+1}''}^{\mathrm{L},\top}$, and where the factors $\dot{A}_i$, A_i', and A_i'' are defined in the same way as in (10).
- The unfoldings $X^{<i>}$, defined in (6), of a TTD $X = X_1 \cdots X_d$ can be rewritten as:

$$X^{<i>} = (X_1 \cdots X_i)^{\mathrm{R}} (X_{i+1} \cdots X_d)^{\mathrm{L}}. \tag{15}$$

From (6) and (15) it can be deduced that

$$r_i = \mathrm{rank}\,(X_1 \cdots X_i)^{\mathrm{R}} = \mathrm{rank}\,(X_{i+1} \cdots X_d)^{\mathrm{L}}, \tag{16}$$

which can be used to prove that the left and right unfolding of each factor X_i has full rank r_{i-1} and r_i respectively.
- Orthogonality between TTDs is exploited frequently in the rest of this paper, e.g., in the parametrization of the tangent cone discussed in Section 2.4. Orthogonality is defined with respect to the inner product (3). Remark that the vectorization of an unfolding is equal to that of the tensor. Thus, if $Y = Y_1 \cdots Y_d$ and $Z = Z_1 \cdots Z_d$, and by using (15), the inner product $\langle Y, Z\rangle$ is zero if at least one of the following equalities holds:

$$(Y_1 \cdots Y_i)^{\mathrm{R},\top} (Z_1 \cdots Z_i)^{\mathrm{R}} = 0, \qquad (Y_{i+1} \cdots Y_d)^{\mathrm{L}} (Z_{i+1} \cdots Z_d)^{\mathrm{L},\top} = 0, \tag{17}$$

for $i = 1,\dots,d-1$.

2.3 | Compact notations

To simplify the expressions in the rest of the paper we introduce a compact notation for the reshape operator of a higher-order tensor to a third-order tensor:

$$Y^{<i,j>} := [Y]^{n_1\cdots n_i \times n_{i+1}\cdots n_j \times n_{j+1}\cdots n_d} := \mathrm{reshape}\,(Y, n_1\cdots n_i \times n_{i+1}\cdots n_j \times n_{j+1}\cdots n_d), \qquad 1 \leq i \leq j \leq d.$$

Remark that if $i = j$ or $j = d$ this notation is equal to $Y^{<i>}$ defined in (6).

Additionally, we define

$$X_{i:j} := X_i \cdots X_j \in \mathbb{R}^{r_{i-1} \times n_i \times \cdots \times n_j \times r_j}, \qquad 1 \leq i < j \leq d,$$

Similarly,

$$\begin{aligned} X'_{i:j} &:= X'_i \cdots X'_j \in \mathbb{R}^{r_{i-1} \times n_i \times \cdots \times n_j \times r_j}, && 1 \leq i < j \leq d-1, \\ X''_{i:j} &:= X''_i \cdots X''_j \in \mathbb{R}^{r_{i-1} \times n_i \times \cdots \times n_j \times r_j}, && 2 \leq i < j \leq d. \end{aligned}$$

Remark that $\left(X''_{i:j}\right)^{\mathrm{L},\top} \in \mathrm{St}\left(r_{i-1}, n_i \cdots n_j r_j\right)$ and $\left(X'_{i:j}\right)^{\mathrm{R}} \in \mathrm{St}\left(r_j, r_{i-1} n_i \cdots n_j\right)$ for any i and j that satisfy the above bounds.

2.4 | Tangent cone

In the next lemma, we recall the parametrization of the tangent cone to the variety of TTDs of bounded TT-rank [18] with modified orthogonality conditions. The orthogonality conditions differ from the ones in [18, Theorem 2.6] in the sense that only orthogonalized factors of the TTD of X are used, i.e., X'_i and X''_i. As a consequence, no matrix inverses are needed for the projections defined in Section 3.1 to make the overall RRAM proposed in Section 4 more stable. A proof is included in the following lemma. A visualization of the parametrization of the tangent cone for a third-order tensor is given in Figure 3.

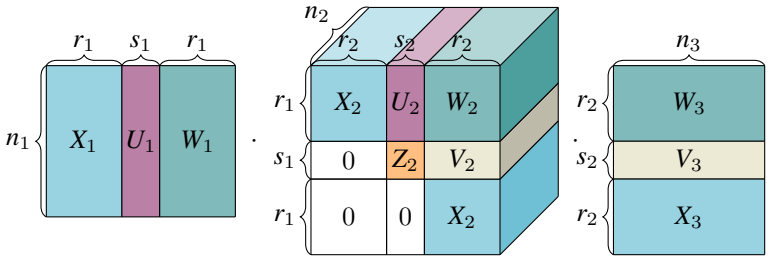

FIGURE 3 Visualization of the parametrization of a tensor in the tangent cone to the variety of bounded TT-rank third-order tensors.

Lemma 1. *Let $X \in \mathbb{R}^{n_1 \times \cdots \times n_d}_{(r_1,\ldots,r_{d-1})}$ as in (10). Then, $T_X \mathbb{R}^{n_1 \times \cdots \times n_d}_{\leq (k_1,\ldots,k_{d-1})}$ is the set of all tensors $G \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ that can be decomposed as*

$$G = \begin{bmatrix} X'_1 & U_1 & W_1 \end{bmatrix} \cdot \begin{bmatrix} X'_2 & U_2 & W_2 \\ 0 & Z_2 & V_2 \\ 0 & 0 & X''_2 \end{bmatrix} \cdots \begin{bmatrix} X'_{d-1} & U_{d-1} & W_{d-1} \\ 0 & Z_{d-1} & V_{d-1} \\ 0 & 0 & X''_{d-1} \end{bmatrix} \cdot \begin{bmatrix} W_d \\ V_d \\ X''_d \end{bmatrix}, \tag{18}$$

where $U_i \in \mathbb{R}^{r_{i-1} \times n_i \times s_i}$, $s_i := k_i - r_i$, for $i = 1, \ldots, d-1$, $W_i \in \mathbb{R}^{r_{i-1} \times n_i \times r_i}$, for $i = 1, \ldots, d$, $Z_i \in \mathbb{R}^{s_{i-1} \times n_i \times s_i}$, for $i = 2, \ldots, d-1$, $V_i \in \mathbb{R}^{s_{i-1} \times n_i \times r_i}$, for $i = 2, \ldots, d$, and

$$\begin{aligned} U_i^{\mathrm{R},\top} X_i'^{\mathrm{R}} &= 0, & W_i^{\mathrm{R},\top} X_i'^{\mathrm{R}} &= 0, & & i = 1, \ldots, d-1, \\ V_i^{\mathrm{L}} (X_i'')^{\mathrm{L},\top} &= 0, & & & & i = 2, \ldots, d. \end{aligned} \tag{19}$$

Proof. $T_X \mathbb{R}^{n_1 \times \cdots \times n_d}_{\leq (k_1,\ldots,k_{d-1})}$ is the set of all $G \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ that can be decomposed as [18, Theorem 2.6]

$$G = \begin{bmatrix} X'_1 & U_1 & \dot{W}_1 \end{bmatrix} \cdot \begin{bmatrix} X'_2 & U_2 & \dot{W}_2 \\ 0 & Z_2 & \dot{V}_2 \\ 0 & 0 & X'_2 \end{bmatrix} \cdots \begin{bmatrix} X'_{d-1} & U_{d-1} & \dot{W}_{d-1} \\ 0 & Z_{d-1} & \dot{V}_{d-1} \\ 0 & 0 & X'_{d-1} \end{bmatrix} \cdot \begin{bmatrix} W_d \\ V_d \\ \dot{X}_d \end{bmatrix},$$

with the orthogonality conditions [18, Theorem 2.6]

$$\begin{aligned} \left(U_i^{\mathrm{R}}\right)^\top X_i'^{\mathrm{R}} &= 0, & \left(\dot{W}_i^{\mathrm{R}}\right)^\top X_i'^{\mathrm{R}} &= 0, & i &= 1,\dots,d-1, \\ \left(\dot{V}_i \cdot X'_{(i+1):(d-1)} \cdot \dot{X}_d\right)^{\mathrm{L}} \left(\left(X'_{i:(d-1)} \cdot \dot{X}_d\right)^{\mathrm{L}}\right)^\top &= 0, & & & i &= 2,\dots,d. \end{aligned} \tag{20}$$

The following invariances hold for all invertible matrices B_i in $\mathrm{GL}(r_i)$, for $i = 1,\dots,d-1$,

$$G = \begin{bmatrix} X'_1 & U_1 & \dot{W}_1 B_1 \end{bmatrix} \cdot \begin{bmatrix} X'_2 & U_2 & \dot{W}_2 \cdot B_2 \\ 0 & Z_2 & \dot{V}_2 \cdot B_2 \\ 0 & 0 & B_1^{-1} \cdot X'_2 \cdot B_2 \end{bmatrix} \cdots \begin{bmatrix} X'_{d-1} & U_{d-1} & \dot{W}_{d-1} \cdot B_{d-1} \\ 0 & Z_{d-1} & \dot{V}_{d-1} \cdot B_{d-1} \\ 0 & 0 & B_{d-2}^{-1} \cdot X'_{d-1} \cdot B_{d-1} \end{bmatrix} \cdot \begin{bmatrix} W_d \\ V_d \\ B_{d-1}^{-1} \dot{X}_d \end{bmatrix}.$$

Because this holds for all invertible B_i, it also holds for the matrices Q_i defined in (12), and thus because of (13), $X_i'' := Q_{i-1}^{-1} \cdot X_i' \cdot Q_i$, for $i = 2,\dots,d$, where $Q_d := I_{n_d}$. Then, we define $W_i := \dot{W}_i Q_i$ and $V_i := V_i Q_i$, for $i = 1,\dots,d-1$, and obtain the parametrization (18). The second set of orthogonality conditions in (20) changes to

$$\left(X_i'^{\mathrm{R}}\right)^\top \dot{W}_i^{\mathrm{R}} = 0 \Leftrightarrow \left(X_i'^{\mathrm{R}}\right)^\top \dot{W}_i^{\mathrm{R}} Q_i = 0 \Leftrightarrow \left(X_i'^{\mathrm{R}}\right)^\top W_i^{\mathrm{R}} = 0, \qquad i = 1,\dots,d,$$

and the third changes to

$$\begin{aligned} \left(\dot{V}_i \cdot X'_{(i+1):(d-1)} \cdot \dot{X}_d\right)^{\mathrm{L}} \left(\left(X'_{i:(d-1)} \cdot \dot{X}_d\right)^{\mathrm{L}}\right)^\top = 0 &\Leftrightarrow \left(\dot{V}_i \cdot Q_i \cdot X''_{(i+1):d}\right)^{\mathrm{L}} \left(\left(Q_{i-1} \cdot X''_{i:d}\right)^{\mathrm{L}}\right)^\top = 0 \\ &\Leftrightarrow \left(V_i \cdot X''_{(i+1):d}\right)^{\mathrm{L}} \left(\left(X''_{i:d}\right)^{\mathrm{L}}\right)^\top Q_{i-1}^\top = 0 \\ &\Leftrightarrow V_i^{\mathrm{L}} \left(\left(X''_{(i+1):d}\right)^{\mathrm{L}} \otimes I_{n_i}\right) \left(\left(X''_{(i+1):d}\right)^{\mathrm{L},\top} \otimes I_{n_i}\right) \left(X_i''\right)^{\mathrm{L},\top} = 0 \\ &\Leftrightarrow V_i^{\mathrm{L}} \left(X_i''\right)^{\mathrm{L},\top} = 0, \end{aligned}$$

for $i = 2,\dots,d$. The first set of orthogonality conditions in (20) is left unchanged. □

Thus, when (18) is expanded, a sum of $\frac{d(d+1)}{2}$ mutual orthogonal terms is obtained:

$$\begin{aligned} G = X'_{1:(d-1)} \cdot W_d + \cdots + W_1 \cdot X''_{2:d} + \\ X'_{1:(d-2)} \cdot U_{d-1} \cdot V_d + \cdots + U_1 \cdot V_2 \cdot X''_{3:d} + \\ X'_{1:(d-3)} \cdot U_{d-2} \cdot Z_{d-1} \cdot V_d + \cdots + U_1 \cdot Z_2 \cdot V_3 \cdot X''_{4:d} + \\ \vdots \\ + U_1 \cdot Z_2 \cdots Z_{d-1} \cdot V_d. \end{aligned} \tag{21}$$

The tangent space is then defined as

$$T_X \mathbb{R}^{n_1 \times \cdots \times n_d}_{(r_1,\dots r_{d-1})} := \left\{ X'_{1:(d-1)} \cdot W_d + \cdots + W_1 \cdot X''_{2:d} \;\middle|\; \text{s.t. } W_i^{\mathrm{R},\top} X_i^{\mathrm{R}} = 0,\ i = 1,\dots,d-1 \right\}. \tag{22}$$

Remark that the orthogonality conditions of the tangent space can easily be changed to

$$\begin{aligned} W_i^{\mathrm{R},\top} X_i'^{\mathrm{R}} &= 0, & i &= 1,\dots,j-1, \\ W_i^{\mathrm{L}} X_i''^{\mathrm{L},\top} &= 0, & i &= j+1,\dots,d, \end{aligned} \tag{23}$$

for any $j \in \{1,\dots,d\}$. For example, we can decompose W_d in (22) as $W_d = \dot{W}_d X_d'' + \hat{W}_d$, where $\hat{W}_d := W_d \left(I - X_d''^\top X_d''\right)$ and $\dot{W}_d := W_d X_d''^\top$. Consequently $X_d'' \hat{W}_d^\top = 0$, and we can regroup the terms involving W_{d-1} and W_d as:

$$X'_{1:(d-2)} \cdot W_{d-1} \cdot X_d'' + X'_{1:(d-1)} \cdot W_d = X'_{1:(d-2)} \cdot \left(W_{d-1} + X'_{d-1} \cdot \dot{W}_d\right) \cdot X_d'' + X'_{1:(d-1)} \cdot \hat{W}_d.$$

And thus, if we define $\tilde{W}_{d-1} := W_{d-1} + X'_{d-1} \cdot \dot{W}_d$, then the modified parameterization of the tangent space with parameters $W_1, \ldots, W_{d-2}, \tilde{W}_{d-1}, \hat{W}_d$ satisfies (23) for $j = d-1$. This process can be applied recursively to obtain (23) for any j.

As for third-order tensors, the projection onto the tangent space is easy and well-known [26, 21]. For the parametrization in Lemma 1, the parameters W_i, $i = 1, \ldots, d$, of $\mathcal{P}_{T_X \mathbb{R}^{n_1 \times \cdots \times n_d}_{(r_1, \ldots, r_{d-1})}} Y$ with $Y \in \mathbb{R}^{n_1 \times \cdots n_d}$ are:

$$\begin{aligned} W_i^{\mathrm{R}} &= P^{\perp}_{X_i'^{\mathrm{R}}} \left(\left(X'_{1:(i-1)} \right)^{\mathrm{R},\top} \cdot Y^{<i>} \cdot \left(X''_{(i+1):d} \right)^{\mathrm{L},\top} \right)^{\mathrm{R}}, && i = 1, \ldots, d-1 \\ W_d &= \left(X'_{1:(d-1)} \right)^{\mathrm{R},\top} Y^{\mathrm{R}}. \end{aligned} \tag{24}$$

Or using the orthogonality conditions (23):

$$\begin{aligned} W_i^{\mathrm{R}} &= P^{\perp}_{X_i'^{\mathrm{R}}} \left(\left(X'_{1:(i-1)} \right)^{\mathrm{R},\top} \cdot Y^{<i>} \cdot \left(X''_{(i+1):d} \right)^{\mathrm{L},\top} \right)^{\mathrm{R}}, && i = 1, \ldots, j-1, \\ W_j &= \left(X'_{1:(j-1)} \right)^{\mathrm{R},\top} \cdot Y^{<j>} \cdot \left(X''_{(j+1):d} \right)^{\mathrm{L},\top}, \\ W_i^{\mathrm{L}} &= \left(\left(X'_{1:(i-1)} \right)^{\mathrm{R},\top} \cdot Y^{<i>} \cdot \left(X''_{(i+1):d} \right)^{\mathrm{L},\top} \right)^{\mathrm{L}} P^{\perp}_{X_i''^{\mathrm{L},\top}}, && i = j+1, \ldots, d. \end{aligned} \tag{25}$$

2.5 | Low-rank tensor approximation problem

In this subsection, we define an auxiliary problem that is related to the tensor completion problem. This auxiliary problem is used in Sections 3.1.2 and 3.2. More specifically, we define the well-known *low-rank tensor approximation problem (LRTAP)* as:

$$\min_{X \in \mathbb{R}^{n_1 \times \cdots \times n_d}_{\leq (r_1, \ldots, r_{d-1})}} \underbrace{\frac{1}{2} \|X - A\|^2}_{=: f(X)}, \qquad A \in \mathbb{R}^{n_1 \times \cdots \times n_d}, \tag{26}$$

This problem is related to the LRTCP (1) because $f_\Omega(X) = f(X)$ for $\Omega = \{1, \ldots, n_1\} \times \cdots \times \{1, \ldots, n_d\}$. Remark that, as for (1), a global minimizer is not unique because of the non-uniqueness of a TTD. Furthermore, in general, a minimizer is NP-hard to obtain [11].

2.6 | Approximate projection

To develop an RRAM, we need to reduce the rank when the iterate comes close to a lower-rank set. Ideally, we want to solve:

$$\mathcal{P}_{\mathbb{R}^{n_1 \times \cdots \times n_d}_{\leq (r_1, \ldots, r_{d-1})}} Y := \operatorname*{argmin}_{X \in \mathbb{R}^{n_1 \times \cdots \times n_d}_{\leq (r_1, \ldots, r_{d-1})}} \|Y - X\|^2, \qquad Y \in \mathbb{R}^{n_1 \times \cdots \times n_d}_{(r'_1, \ldots, r'_{d-1})}, \tag{27}$$

where $\|\cdot\|$ is again the norm induced by (3) and $r_i \leq r'_i$, for $i = 1, \ldots, d-1$. Since the low-rank variety is a closed cone, it holds that [20, Proposition A.6]

$$\left\langle Y - \hat{Y}, \hat{Y} \right\rangle = 0 \Leftrightarrow \left\langle Y, \hat{Y} \right\rangle = \left\| \hat{Y} \right\|^2, \tag{28}$$

for all $Y \in \mathbb{R}^{n_1\times\cdots\times n_d}_{(r'_1,\ldots,r'_{d-1})}$ and $\hat{Y} \in \mathcal{P}_{\mathbb{R}^{n_1\times\cdots\times n_d}_{\leq(r_1,\ldots,r_{d-1})}} Y$. Thus, (27) can be rewritten as

$$\mathcal{P}_{\mathbb{R}^{n_1\times\cdots\times n_d}_{\leq(r_1,\ldots,r_{d-1})}} Y = \operatorname*{argmin}_{\substack{X\in\mathbb{R}^{n_1\times\cdots\times n_d}_{\leq(r_1,\ldots,r_{d-1})}\\ \langle Y,X\rangle=\|X\|^2}} \|Y-X\|^2 = \operatorname*{argmin}_{\substack{X\in\mathbb{R}^{n_1\times\cdots\times n_d}_{\leq(r_1,\ldots,r_{d-1})}\\ \langle Y,X\rangle=\|X\|^2}} -2\langle Y,X\rangle + \|X\|^2 = \operatorname*{argmax}_{\substack{X\in\mathbb{R}^{n_1\times\cdots\times n_d}_{\leq(r_1,\ldots,r_{d-1})}\\ \langle Y,X\rangle=\|X\|^2}} \|X\|$$
$$= \operatorname*{argmax}_{\substack{X\in\mathbb{R}^{n_1\times\cdots\times n_d}_{\leq(r_1,\ldots,r_{d-1})}\\ \langle Y,X\rangle=\|X\|^2}} \left\langle Y, \frac{X}{\|X\|}\right\rangle,$$

or

$$\left\|\hat{Y}\right\| = \max_{\substack{X\in\mathbb{R}^{n_1\times\cdots\times n_d}_{\leq(r_1,\ldots,r_{d-1})}\\ \langle Y,X\rangle=\|X\|^2}} \left\langle Y, \frac{X}{\|X\|}\right\rangle = \max_{\substack{X\in\mathbb{R}^{n_1\times\cdots\times n_d}_{\leq(r_1,\ldots,r_{d-1})}\\ \langle Y,X\rangle=\|X\|^2}} \|X\|, \tag{29}$$

for all $\hat{Y} \in \mathcal{P}_{\mathbb{R}^{n_1\times\cdots\times n_d}_{\leq(r_1,\ldots,r_{d-1})}} Y$, and thus all elements of $\mathcal{P}_{\mathbb{R}^{n_1\times\cdots\times n_d}_{\leq(r_1,\ldots,r_{d-1})}} Y$ have the same norm. However, as both varieties are highly non-convex, solving (29) is non-trivial. That is why we search for an *approximate projection*, i.e., a set-valued mapping:

$$\tilde{\mathcal{P}}_{\mathbb{R}^{n_1\times\cdots\times n_d}_{\leq(r_1,\ldots,r_{d-1})}} : \mathbb{R}^{n_1\times\cdots\times n_d}_{(r'_1,\ldots,r'_{d-1})} \multimap \mathbb{R}^{n_1\times\cdots\times n_d}_{\leq(r_1,\ldots,r_{d-1})},$$

such that there exists $\omega \in (0,1]$ such that, for all $Y \in \mathbb{R}^{n_1\times\cdots\times n_d}_{(r'_1,\ldots,r'_{d-1})}$ and all $\tilde{Y} \in \tilde{\mathcal{P}}_{\mathbb{R}^{n_1\times\cdots\times n_d}_{\leq(r_1,\ldots,r_{d-1})}} Y$,

$$\left\langle Y, \frac{\tilde{Y}}{\|\tilde{Y}\|}\right\rangle \geq \omega \left\|\mathcal{P}_{\mathbb{R}^{n_1\times\cdots\times n_d}_{\leq(r_1,\ldots,r_{d-1})}} Y\right\|. \tag{30}$$

Inequality (30) is called an *angle condition* [24, Definition 2.5]; it is well defined since, as $\mathbb{R}^{n_1\times\cdots\times n_d}_{\leq(r_1,\ldots,r_{d-1})}$ is a closed cone, all elements of $\mathcal{P}_{\mathbb{R}^{n_1\times\cdots\times n_d}_{\leq(r_1,\ldots,r_{d-1})}} Y$ have the same norm. If the approximate projection is chosen such that all $\tilde{Y} \in \tilde{\mathcal{P}}_{\mathbb{R}^{n_1\times\cdots\times n_d}_{\leq(r_1,\ldots,r_{d-1})}} Y$ satisfy $\langle Y,\tilde{Y}\rangle = \|\tilde{Y}\|^2$, then the angle condition simplifies to

$$\|\tilde{Y}\| \geq \omega \left\|\mathcal{P}_{\mathbb{R}^{n_1\times\cdots\times n_d}_{\leq(r_1,\ldots,r_{d-1})}} Y\right\|. \tag{31}$$

We use this simplified angle condition in Section 3.2.

A visualization of an approximate projection onto a cone is given in Figure 4.

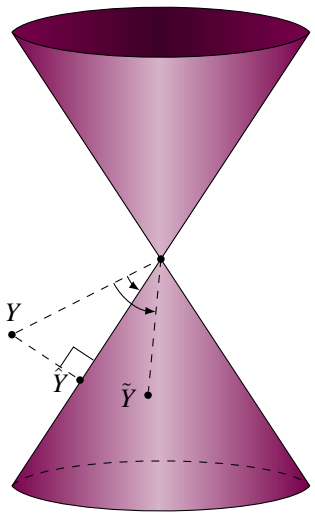

FIGURE 4 Visualization of an approximate projection onto a cone.

3 | METHODOLOGY

This section gives our main contributions to the RRAM – the method to increase and decrease the rank. For both methods, numerical experiments are presented to illustrate the use and added value of the methods.

3.1 | Rank increase

In Section 3.1.1, we propose a simplified projection onto the tangent cone to determine the search direction. Afterward, in Section 3.1.2 we propose a method to determine an adequate value for the upper bound on the rank for the LRTCP. This involves a theoretical result for the LRTAP. We define an *estimated rank* to extend this result to the LRTCP. Then in Section 3.1.3, some experiments are given to illustrate the use of the rank estimation method. Lastly, in Section 3.1.4 we give the full method that is used in the RRAM to increase the rank.

3.1.1 | Search direction in the tangent cone

Inspired by [7], the proposed RRAM uses a search direction selected in the normal part of the tangent cone with respect to the tangent space, which is the closed cone

$$T_X^{\perp}\mathbb{R}^{n_1\times\cdots\times n_d}_{\leq(k_1,\ldots,k_{d-1})} = T_X\mathbb{R}^{n_1\times\cdots\times n_d}_{\leq(k_1,\ldots,k_{d-1})} \cap \left(T_X\mathbb{R}^{n_1\times\cdots\times n_d}_{(r_1,\ldots,r_{d-1})}\right)^{\perp},$$

to increase the rank. However, as can be seen from (21), the terms in the normal part of the tangent cone are highly non-linear and non-convex. In [33], an approximate projection onto the tangent cone for third-order tensors was proposed. However, for higher-order tensors, such an approximate projection gets increasingly complex and computationally expensive. In [19], an approximate projection was proposed based on the maximal absolute value in the tensor. However, the angle condition that is obtained decreases exponentially with the order of the tensor: $\omega = \frac{1}{\frac{d(d^2-1)}{2}\sqrt{n_1\cdots n_d}}$. As a consequence, we propose to project iteratively onto the subcones

$$T_X^{\perp}\mathbb{R}^{n_1\times\cdots\times n_d}_{\leq(r_1,\ldots,r_{i-1},k_i,r_{i+1},\ldots,r_{d-1})} = \left\{ X'_{1:(i-1)}\cdot U_i\cdot V_{i+1}\cdot X''_{(i+2):d} \;\middle|\; \text{s.t. } U_i^{\mathrm{R},\top}X_i'^{\mathrm{R}} = 0,\ V_{i+1}^{\mathrm{L}}X_{i+1}''^{\,\mathrm{L},\top} = 0 \right\},$$

for $i = 1,\ldots,d-1$. These are the terms on the second line in (21), i.e., the expanded version of the parametrization of the tangent cone in (18). This tangent cone is written more shortly as $T_X^{\perp}\mathbb{R}^{n_1\times\cdots\times n_d}_{\leq(r_{1:(i-1)},k_i,r_{(i+1):d})}$. The projection is much easier as each element is only a second-order function in the parameters. The projection is given in the following proposition.

Proposition 1. *Let $X \in \mathbb{R}^{n_1\times\cdots\times n_d}_{\leq(r_1,\ldots,r_{d-1})}$ as in (10) and $Y \in \mathbb{R}^{n_1\times\cdots\times n_d}$. A projection onto $T_X^{\perp}\mathbb{R}^{n_1\times\cdots\times n_d}_{\leq(r_{1:(i-1)},k_i,r_{(i+1):d})}$:*

$$\mathcal{P}_{T_X^{\perp}\mathbb{R}^{n_1\times\cdots\times n_d}_{\leq(r_{1:(i-1)},k_i,r_{(i+1):d})}} : \mathbb{R}^{n_1\times\cdots\times n_d} \multimap T_X^{\perp}\mathbb{R}^{n_1\times\cdots\times n_d}_{\leq(r_{1:(i-1)},k_i,r_{(i+1):d})} : Y \to \hat{Y},$$

where $i \in \{1,\ldots,d-1\}$, and

$$\hat{Y} = X'_{1:(i-1)}\cdot U_i\cdot V_{i+1}\cdot X''_{(i+2):d} \in \mathcal{P}_{T_X^{\perp}\mathbb{R}^{n_1\times\cdots\times n_d}_{\leq(r_{1:(i-1)},k_i,r_{(i+1):d})}} Y,$$

is given by

$$\left[U_i^{\mathrm{R}}, S, V\right] = \mathrm{SVD}_{s_i}\left(P_i\left(X,Y\right)\right), \qquad V_{i+1}^{\mathrm{L}} = SV^{\top}, \tag{32}$$

where SVD_{s_i} denotes a truncated SVD at rank s_i, and $s_i := k_i - r_i$, and

$$P_i(X,Y) := P^{\perp}_{X_i'^{\mathrm{R}}}\left[\left(X'_{1:(i-1)}\right)^{\mathrm{R},\top}\cdot Y^{<i-1,i+1>}\cdot\left(X''_{(i+2):d}\right)^{\mathrm{L},\top}\right]^{r_{i-1}n_i\times n_{i+1}r_i} P^{\perp}_{X_{i+1}''^{\,\mathrm{L},\top}}. \tag{33}$$

Proof. Because of (29), $\hat{Y} \in \mathcal{P}_{T_X^{\perp}\mathbb{R}^{n_1\times\cdots\times n_d}_{\leq(r_{1:(i-1)},k_i,r_{(i+1):d})}}$ maximizes the following inner product

$$\left\{U_i, V_{i+1}\right\} = \underset{U,V}{\operatorname{argmin}} \left\langle Y, X'_{1:(i-1)} \cdot U \cdot V \cdot X''_{(i+2):d} \right\rangle \quad \text{s.t.} \quad \left\langle Y, X'_{1:(i-1)} \cdot U \cdot V \cdot X''_{(i+2):d} \right\rangle = \left\| X'_{1:(i-1)} \cdot U \cdot V \cdot X''_{(i+2):d} \right\|^2,$$

for all $X \in \mathbb{R}^{n_1\times\cdots\times n_d}_{\leq(r_1,\ldots,r_{d-1})}$. Furthermore, any tensor Y can be decomposed as

$$Y^{<i-1,i+1>} = P_{\left(X'_{1:(i-1)}\right)^{\mathrm{R}}} Y^{<i-1,i+1>} P_{\left(X''_{(i+2):d}\right)^{\mathrm{L},\top}} + P^{\perp}_{\left(X'_{1:(i-1)}\right)^{\mathrm{R}}} Y^{<i-1,i+1>} P_{\left(X''_{(i+2):d}\right)^{\mathrm{L},\top}} + Y^{<i-1,i+1>} P^{\perp}_{\left(X''_{(i+2):d}\right)^{\mathrm{L},\top}}. \tag{34}$$

The first term can further be decomposed as

$$\begin{aligned}
&P_{\left(X'_{1:(i-1)}\right)^{\mathrm{R}}} Y^{<i-1,i+1>} P_{\left(X''_{(i+2):d}\right)^{\mathrm{L},\top}} = \\
&\left(X'_{1:(i-1)}\right)^{\mathrm{R}} \Bigg[P^{\perp}_{X'^{\mathrm{R}}_i} \Big[\left(X'_{1:(i-1)}\right)^{\mathrm{R},\top} Y^{<i-1,i+1>} \left(X''_{(i+2):d}\right)^{\mathrm{L},\top} \Big]^{r_{i-1}n_i\times n_{i+1}r_{i+1}} P^{\perp}_{X''^{\mathrm{L},\top}_{i+1}} \\
&\quad + P_{X'^{\mathrm{R}}_i} \Big[\left(X'_{1:(i-1)}\right)^{\mathrm{R},\top} Y^{<i-1,i+1>} \left(X''_{(i+2):d}\right)^{\mathrm{L},\top} \Big]^{r_{i-1}n_i\times n_{i+1}r_{i+1}} P^{\perp}_{X''^{\mathrm{L},\top}_{i+1}} \\
&\quad + \Big[\left(X'_{1:(i-1)}\right)^{\mathrm{R},\top} Y^{<i-1,i+1>} \left(X''_{(i+2):d}\right)^{\mathrm{L},\top} \Big]^{r_{i-1}n_i\times n_{i+1}r_{i+1}} P_{X''^{\mathrm{L},\top}_{i+1}} \Bigg]^{r_{i-1}\times n_i n_{i+1}\times r_{i+i}} \left(X''_{(i+2):d}\right)^{\mathrm{L}},
\end{aligned}$$

which is a sum of three mutually orthogonal terms. If we insert this in (34), Y can be written as the sum of five mutually orthogonal terms. Thus, the inner product with $\hat{Y} = X'_{1:(i-1)} \cdot U_i \cdot V_{i+1} \cdot X''_{(i+2):d}$ is

$$\left\langle \hat{Y}, Y \right\rangle = \left\langle \hat{Y}^{<i-1,i+1>}, \left(X'_{1:(i-1)}\right)^{\mathrm{R}} P_i(X,Y) \left(X''_{(i+2):d}\right)^{\mathrm{L},\top} \right\rangle = \left\langle \left[U_i \cdot V_{i+1}\right]^{r_{i-1}n_i\times n_{i+1}r_i}, P_i(X,Y) \right\rangle,$$

where we used the fact that the multiplication with the orthogonal matrices $\left(X'_{1:(i-1)}\right)^{\mathrm{R}}$ and $\left(X''_{(i+2):d}\right)^{\mathrm{L},\top}$ on the left and right of both terms does not change the inner product. Because $\left[U_i \cdot V_{i+1}\right]^{r_{i-1}n_i\times n_{i+1}r_i}$ is just a matrix of rank s_i, the optimal U_i and V_{i+1} can for example be obtained by the truncated SVD of $P_i(X,Y)$ as in (32). □

3.1.2 | Rank estimation

In this subsection, a method to estimate how much the rank should be increased is proposed. This method is a generalization of the rank estimation method proposed for third-order tensors in [34]. First, a theoretical result for the LRTAP is given and afterwards, we discuss how we use this result in the LRTCP.

Theorem 1. *Let $X \in \mathbb{R}^{n_1\times\cdots\times n_d}_{(r_1,\ldots,r_{d-1})}$ and $A \in \mathbb{R}^{n_1\times\cdots\times n_d}_{(r'_1,\ldots,r'_{d-1})}$ with orthogonalizations as in (10). If $\mathcal{P}_{T_X\mathbb{R}^{n_1\times\cdots\times n_d}_{(r_1,\ldots,r_{d-1})}} \nabla f(X) = 0$, then*

$$\dot{X}_i = B_i^{\top} \cdot \dot{A}_i \cdot C_i, \qquad i = 1,\ldots,d, \tag{35}$$

where $B_1 = I_{n_1}$, $C_d = I_{n_d}$, and

$$B_i^{\top} = \left(X'_{1:(i-1)}\right)^{\mathrm{R},\top} \left(A'_{1:(i-1)}\right)^{\mathrm{R}}, \qquad C_i = \left(A''_{(i+1):d}\right)^{\mathrm{L}} \left(X''_{(i+1):d}\right)^{\mathrm{L},\top}.$$

And using (12) and (14):

$$\begin{aligned}
X'_i &= B_i^{\top} \cdot A'_i \cdot Q'_i C_i Q_i^{-1}, && i = 1,\ldots,d-1, \\
X''_i &= Q_{i-1}^{-1} B_i^{\top} Q'_{i-1} \cdot A''_i \cdot C_i, && i = 2,\ldots,d.
\end{aligned} \tag{36}$$

Thus, if we define the matrices

$$D_i := C_i Q_i^{-1} B_{i+1}^{\top} Q'_i, \qquad E_i := Q'_i C_i Q_i^{-1} B_{i+1}^{\top}, \qquad i = 1,\ldots,d-1,$$

X can be written as a function of the factors of A as follows

$$X = A'_1 E_1 \cdots A'_{i-1} \cdot E_{i-1} \cdot \dot{A}_i \cdot D_i \cdot A''_{i+1} \cdots D_{d-1} A''_d, \qquad i \in \{1, \ldots, d\}.$$

Furthermore, $B_2 = Q'_1 C_1 Q_1^{-1} \in \mathrm{St}(r_1, r'_1)$, $C_{d-1} = \left(Q_{d-1}^{-1} B_d^\top Q'_{d-1}\right)^\top \in \mathrm{St}(r_{d-1}, r'_{d-1})$, and D_i and E_i have r_i times eigenvalue 1 and the other $r'_i - r_i$ eigenvalues are zero for all i. More specifically,

$$B_{i+1}^\top Q'_i C_i Q_i^{-1} = Q_i^{-1} B_{i+1}^\top Q'_i C_i = I_{r_i}, \tag{37}$$

and thus $B_{i+1}^\top Q'_i$ and $C_i Q_i^{-1}$ are respectively right and left eigenvectors of D_i corresponding to eigenvalue 1 and the same holds for $Q_i^{-1} B_{i+1}^\top$, $Q'_i C_i$, and E_i.

Proof. From (24), and because $\mathcal{P}_{T_X \mathbb{R}^{n_1 \times \cdots \times n_d}_{(r_1, \ldots, r_{d-1})}} \nabla f(X) = 0$:

$$\begin{aligned} P^\perp_{X_i'^{\mathrm{R}}} \left(\left(X'_{1:(i-1)}\right)^{\mathrm{R},\top} \cdot A^{<i>} \cdot \left(X''_{(i+1):d}\right)^{\mathrm{L},\top} \right)^{\mathrm{R}} &= 0, \qquad i = 1, \ldots, d-1 \\ \dot{X}_d - \left(X'_{1:(d-1)}\right)^{\mathrm{R},\top} A^{\mathrm{R}} &= 0. \end{aligned} \tag{38}$$

Using the last equation, we have:

$$\dot{X}_d = \left(X'_{1:(d-1)}\right)^{\mathrm{R},\top} \left(A'_{1:(d-1)}\right)^{\mathrm{R}} \dot{A}_d.$$

The factor $B_d^\top := \left(X'_{1:(d-1)}\right)^{\mathrm{R},\top} \left(A'_{1:(d-1)}\right)^{\mathrm{R}}$ is a full rank matrix of dimension $r_{d-1} \times r'_{d-1}$ and using (12) and (14):

$$X''_d = Q_{d-1}^{-1} B_d^\top Q'_{d-1} A''_d, \tag{39}$$

satisfying (36) and because X''_d are A''_d are orthogonal matrices, it holds that $\left(Q_{d-1}^{-1} B_d^\top Q'_{d-1}\right)^\top \in \mathrm{St}(r_{d-1}, r'_{d-1})$.

Now, we make use of the fact that for any of the orthogonality conditions in (23), the parameters W_j in (25) have to be zero. Suppose that $j = d-1$, then

$$\dot{X}_{d-1} - \left(X'_{1:(d-2)}\right)^{\mathrm{R},\top} \left(A'_{1:(d-2)}\right)^{\mathrm{R}} \cdot \dot{A}_{d-1} \cdot A''_d X''^\top_d = 0,$$

where again the factor $B_{d-1}^\top := \left(X'_{1:(d-2)}\right)^{\mathrm{R},\top} \left(A'_{1:(d-2)}\right)^{\mathrm{R}}$ is a full rank matrix of size $r_{d-2} \times r'_{d-2}$. And using (39), this equation can be simplified to

$$\dot{X}_{d-1} = B_{d-1}^\top \cdot \dot{A}_{d-1} \cdot \left(Q_{d-1}^{-1} B_d^\top Q'_{d-1}\right)^\top.$$

Thus, $C_{d-1} = \left(Q_{d-1}^{-1} B_d^\top Q'_{d-1}\right)^\top$ and since $\left(Q_{d-1}^{-1} B_d^\top Q'_{d-1}\right)^\top \in \mathrm{St}(r_{d-1}, r'_{d-1})$, $C_{d-1} Q_{d-1}^{-1} B_d^\top Q'_{d-1}$ has indeed r_{d-1} times eigenvalue 1 and $r'_{d-1} - r_{d-1}$ times eigenvalue zero. Also for the other values of j, it directly follows that (35) holds by setting W_j to zero in (25).

To prove that D_i and E_i have r_i times eigenvalue 1 for all i, we use the fact that

$$\begin{aligned} B_i^\top = {X'_{i-1}}^{\mathrm{R},\top} \left(I_{n_{i-1}} \otimes \left(X'_{1:(i-2)}\right)^{\mathrm{R},\top}\right) \left(I_{n_{i-1}} \otimes \left(A'_{1:(i-2)}\right)^{\mathrm{R}}\right) {A'_{i-1}}^{\mathrm{R}} &= {X'_{i-1}}^{\mathrm{R},\top} \left(I_{n_{i-1}} \otimes B_{i-1}^\top\right) {A'_{i-1}}^{\mathrm{R}} \\ &= {X'_{i-1}}^{\mathrm{R},\top} \left(B_{i-1}^\top \cdot A'_{i-1}\right)^{\mathrm{R}}. \end{aligned} \tag{40}$$

And thus,

$$D_{i-1} = C_{i-1} Q_{i-1}^{-1} {X'_{i-1}}^{\mathrm{R},\top} \left(B_{i-1}^\top \cdot A'_{i-1}\right)^{\mathrm{R}} Q'_{i-1}.$$

If we multiply this matrix on the right with $C_{i-1} Q_{i-1}^{-1}$, we obtain

$$D_{i-1} C_{i-1} Q_{i-1}^{-1} = C_{i-1} Q_{i-1}^{-1} {X'_{i-1}}^{\mathrm{R},\top} \left(B_{i-1}^\top \cdot A'_{i-1}\right)^{\mathrm{R}} Q'_{i-1} C_{i-1} Q_{i-1}^{-1} = C_{i-1} Q_{i-1}^{-1} {X'_{i-1}}^{\mathrm{R},\top} {X'_{i-1}}^{\mathrm{R}} = C_{i-1} Q_{i-1}^{-1},$$

and thus $C_{i-1}Q_{i-1}^{-1}$ are r_{i-1} eigenvectors of D_{i-1} with eigenvalue 1. Since the matrix D_{i-1} has rank r_{i-1}, the other eigenvalues have to be zero. Lastly, the eigenvalues of

$$Q'_{i-1}D_{i-1}\left(Q'_{i-1}\right)^{-1} = Q'_{i-1}C_{i-1}Q_{i-1}^{-1}B_i^\top = E_{i-1}$$

are the same as those of D_{i-1}. □

The following proposition gives a formula for the rank difference between $X^{<i>}$ and $A^{<i>}$ in the case where the Riemannian gradient of the LRTAP at X is zero. This rank difference thus enables to determine the TT-rank of A.

Proposition 2. *If the same conditions as in Theorem 1 hold, then*

$$r'_i - r_i = \operatorname{rank}\left(\left[\left(X'_{1:(i-1)}\right)^{\mathrm{R},\top} \cdot \nabla f(X)^{<i-1,i+1>} \cdot \left(X''_{(i+2):d}\right)^{\mathrm{L},\top}\right]^{r_{i-1}n_i \times n_{i+1}r_{i+1}}\right),$$

for $i = 2,\dots,d-2$, and where $r_0 := r_d := 1$.

Proof. Using Theorem 1, it holds that

$$\begin{aligned}
&\operatorname{rank}\left(\left[\left(X'_{1:(i-1)}\right)^{\mathrm{R},\top} \cdot \nabla f(X)^{<i-1,i+1>} \cdot \left(X''_{(i+2):d}\right)^{\mathrm{L},\top}\right]^{r_{i-1}n_i \times n_{i+1}r_{i+1}}\right)\\
&= \operatorname{rank}\left(\left[X'_i \cdot \dot{X}_{i+1} - B_i^\top \cdot A'_i \cdot \dot{A}_{i+1} \cdot C_{i+1}\right]^{r_{i-1}n_i \times n_{i+1}r_{i+1}}\right)\\
&= \operatorname{rank}\left(\left[B_i^\top \cdot A'_i \cdot E_i \cdot \dot{A}_{i+1} \cdot C_{i+1} - B_i^\top \cdot A'_i \cdot \dot{A}_{i+1} \cdot C_{i+1}\right]^{r_{i-1}n_i \times n_{i+1}r_{i+1}}\right)\\
&= \operatorname{rank}\left(\left[B_i^\top \cdot A'_i \cdot \left(E_i - I_{r'_i}\right) \cdot \dot{A}_{i+1} \cdot C_{i+1}\right]^{r_{i-1}n_i \times n_{i+1}r_{i+1}}\right)\\
&= \operatorname{rank}\left(E_i - I_{r'_i}\right) = r'_i - r_i.
\end{aligned}$$

Remark that the last equality holds because $E_iE_i = E_i$ and thus the space spanned by E_i lies in the null space of $\left(E_i - I_{r'_i}\right)$, which reduces the full rank r'_i to $r'_i - r_i$. □

Remark that when Y is the gradient $\nabla f(X)$, and the orthogonality conditions in (25) hold for $j = i+1$, then

$$\begin{aligned}
&\left[\left(X'_{1:(i-1)}\right)^{\mathrm{R},\top} \nabla f(X)^{<i-1,i+1>} \left(X''_{(i+2):d}\right)^{\mathrm{L},\top}\right]^{r_{i-1}n_i \times n_{i+1}r_i} =\\
&P^{\perp}_{X'^{\mathrm{R}}_i}\left[\left(X'_{1:(i-1)}\right)^{\mathrm{R},\top} \nabla f(X)^{<i-1,i+1>} \left(X''_{(i+2):d}\right)^{\mathrm{L},\top}\right]^{r_{i-1}n_i \times n_{i+1}r_i} P^{\perp}_{X''^{\mathrm{L},\top}_{i+1}} +\\
&P_{X'^{\mathrm{R}}_i}\left[\left(X'_{1:(i-1)}\right)^{\mathrm{R},\top} \nabla f(X)^{<i-1,i+1>} \left(X''_{(i+2):d}\right)^{\mathrm{L},\top}\right]^{r_{i-1}n_i \times n_{i+1}r_i} +\\
&P^{\perp}_{X'^{\mathrm{R}}_i}\left[\left(X'_{1:(i-1)}\right)^{\mathrm{R},\top} \nabla f(X)^{<i-1,i+1>} \left(X''_{(i+2):d}\right)^{\mathrm{L},\top}\right]^{r_{i-1}n_i \times n_{i+1}r_i} P_{X''^{\mathrm{L},\top}_{i+1}} = P_i\left(X, \nabla f(X)\right) + X'^{\mathrm{R}}_i W^{\mathrm{L}}_{i+1} + W^{\mathrm{R}}_i X''^{\mathrm{L}}_{i+1}.
\end{aligned}$$

Thus, when the Riemannian gradient is zero, it holds that

$$\left[\left(X'_{1:(i-1)}\right)^{\mathrm{R},\top} \nabla f(X)^{<i-1,i+1>} \left(X''_{(i+2):d}\right)^{\mathrm{L},\top}\right]^{r_{i-1}n_i \times n_{i+1}r_i} = P_i\left(X, \nabla f(X)\right).$$

Consequently, if the gradient is not exactly zero, it might be a good idea in practice to use $P_i(X, \nabla f(X))$ to estimate the rank A instead of the expression proposed in Proposition 2.

To extend the result of Proposition 2 to the LRTCP, we use the following definition of *estimated rank* of a matrix inspired by [7].

Definition 1 (Estimated rank)**.** Given $B \in \mathbb{R}^{m\times n}$ and $s < \text{rank}\,(B)$, the estimated rank is defined as

$$\tilde{r}_s\,(B) := \begin{cases} 0 & \text{if } B = 0 \\ \text{argmax}_{j\leq s} \frac{\sigma_j(B)-\sigma_{j+1}(B)}{\sigma_j(B)} & \text{otherwise} \end{cases}, \tag{41}$$

where $\sigma_j\,(B), j = 1,\ldots,\text{rank}(B)$, denote the singular values of B in decreasing order, i.e., $\sigma_i(B) \geq \sigma_j(B)$ for $i \leq j$.

Remark: In the above definition, the estimated rank can not equal the full rank. This is not a problem because it is implicitly assumed that B is approximately of low rank. The upper bound s prevents the estimated rank from being too high and should be chosen by the user.

To obtain a stationary point on the manifold we optimize:

$$\min_{X\in\mathbb{R}^{n_1\times\cdots\times n_d}_{(r_1,\ldots,r_{d-1})}} \frac{1}{2}\|X_\Omega - A_\Omega\|^2, \tag{42}$$

using a Riemannian conjugate gradient algorithm developed for tensor completion in the TT format [26].

An overview of the method to estimate the rank of a tensor A based on the sparse tensor A_Ω, is given in Algorithm 1. The next subsection shows some numerical experiments using this algorithm.

Algorithm 1 Rank estimation method for the LRTCP (1).

Input: $\Omega \subseteq \{1,\ldots,n_1\}\times\cdots\times\{1,\ldots,n_d\}, A_\Omega, r_i \in \mathbb{N}_0 \;\; \forall i \in \{1,\ldots,d-1\}$

1: $X^* \leftarrow$ (approximate) minimizer of $\min_{X\in\mathbb{R}^{n_1\times\cdots\times n_d}_{(r_1,\ldots,r_{d-1})}} \frac{1}{2}\|X_\Omega - A_\Omega\|^2$;

2: **for** $i = 1,\ldots,d-1$ **do**

3: $\quad k_i \leftarrow r_i + \tilde{r}_s\big(P_i\,(X^*, \nabla f_\Omega(X^*))\,\big)$ (33), (41);

4: **end for**

Output: $(k_1,\ldots,k_{d-1})$

3.1.3 | Experiments rank estimation

In this section, we show the results of some numerical experiments to illustrate the use of Algorithm 1. For these experiments, $A \in \mathbb{R}^{n_1\times\cdots\times n_d}_{(r'_1,\ldots,r'_{d-1})}$ is generated as follows:

$$A = \texttt{randn}\,(n_1 \times r'_1)\cdot\texttt{randn}\,(r'_1\times n_2\times r'_2)\cdots\texttt{randn}\,(r'_{d-2}\times n_{d-1}\times r'_{d-1})\cdot\texttt{randn}\,(r'_{d-1}\times n_d)\,. \tag{43}$$

It can be shown that A, generated in this way, has approximately standard deviation $\sqrt{r'_1\cdots r'_{d-1}}$. To obtain A_Ω, we randomly sample A using `randperm` in Matlab. We solve (42) using a Riemannian CG algorithm [26, 1]. The starting point X_0 was generated randomly in $\mathbb{R}^{n_1\times\cdots\times n_d}_{(r_1,\ldots,r_{d-1})}$ in the same way as A in (43).

The results of the first experiment are shown in Figure 5. For this experiment, we choose $d := 4$, $n_i := 15$, for $i = 1,\ldots,d$, $r'_i := 3$ and $r_i := 1$, for $i = 1,\ldots,d-1$, and $\rho_\Omega = 0.3$. The value of the Riemannian gradient at X^* that is obtained for this experiment is approximately 10^{-11} after 87 iterations. The leftmost column of subfigures shows the first seven singular values of the matrices $P_i\,(X^*, \nabla f_\Omega(X^*))$, for $i = 1,\ldots,d-1$. The second column shows the relative gap between these singular values, which is used to compute the estimated rank. As can be seen, the combination of the estimated ranks of these matrices indeed equals the difference between the TT-rank of A and X^*. On the other hand, in the third and fourth columns, the singular values and relative gap of the unfoldings of A_Ω are respectively shown. As can be seen, based on the estimated rank of these unfoldings, the rank of A can not be estimated. Furthermore, computing the singular values of the unfoldings of A_Ω is computationally more expensive ($\mathcal{O}\left(\max_i(n_i)^d\right)$ instead of $\mathcal{O}\left(\max_i(n_i)^3\right)$). As a reference, in the rightmost column, the singular values of the unfoldings of A are shown to illustrate the true TT-rank of A.

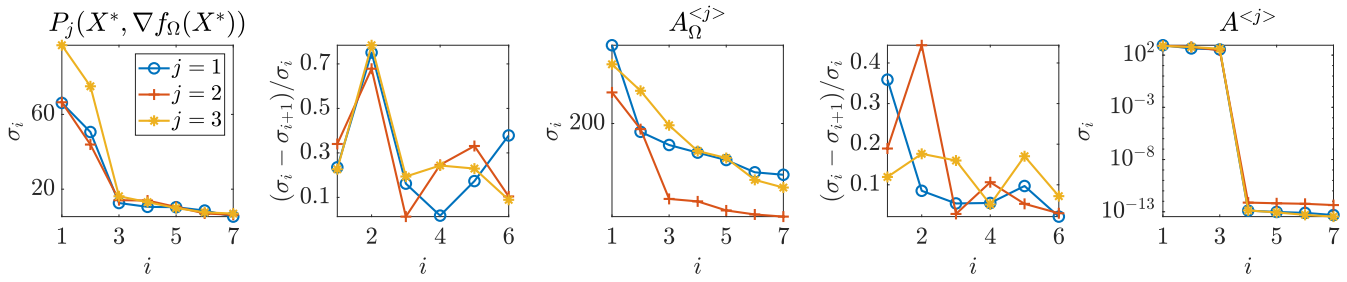

FIGURE 5 *Illustration of the use of Algorithm 1 to determine the rank difference between A and X^*.* The first 7 singular values of the matrices in (33), the relative gap between these singular values, the singular values of the unfoldings of A_Ω and A, where X^* is obtained after 87 iterations of the CG algorithm, and with $d := 4$, $n_i := 15$, for $i = 1, \ldots, d$, $r_i' := 3$ and $r_i = 1$, for $i = 1, \ldots, d-1$, and $\rho_\Omega := 0.3$. The norm of the Riemannian gradient that is obtained at X^* is approximately 10^{-11}.

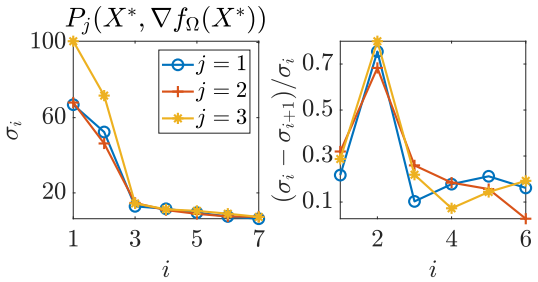

FIGURE 6 *Illustration of the use of Algorithm 1 for approximate minimizers.* The same experiment as in Figure 5 after 15 iterations of the CG algorithm. The norm of the Riemannian gradient that is obtained at X^* is approximately 31.

A second example is shown in Figure 6, where the number of iterations is lowered to 15. The norm of the gradient that is obtained in this case is approximately 31. The results are very similar and the estimated ranks of the matrices $P_i\left(X^*, \nabla f_\Omega(X^*)\right)$ still form the TT-rank of A minus the TT-rank of X^*. A low norm of the gradient and consequently a high number of iterations is in practice thus not necessary in Algorithm 1 to estimate the TT-rank of A. Remark that the singular values of the unfoldings of A_Ω and A have not changed compared to the previous experiment and these subplots are thus omitted.

In a third experiment, the rank of A is changed to $[2, 5, 3]$ and the results are shown in Figure 7. In this figure, only one mode is highlighted in each subfigure. Furthermore, the first seven singular values of the unfoldings of A and A_Ω are shown on the same subplot to illustrate the effect of the sampling more clearly. Furthermore, the relative gap between the singular values for A_Ω is no longer shown because it is of no practical use in the algorithm. The other parameters and maximal number of iterations were the same as in the previous experiment. The norm of the Riemannian gradient that is obtained is approximately 13. Again, the matrices in (33) can be used to determine the TT-rank of A.

In a fourth experiment, the dimension is increased to $d := 5$ and the value of n_i is lowered to 10 for all $i = 1, \ldots, d$. The other parameters were the same as in the second experiment. The CG algorithm is run for 15 iterations and the norm of the Riemannian gradient is approximately 12. The results are shown in Figure 8. Again, Algorithm 1 can determine the exact TT-rank of A based on the sampled elements.

In the last experiment, noise is added to the tensor as follows:

$$A_\eta := A + \eta \, \texttt{randn}\left(n_1 \times \cdots \times n_d\right), \tag{44}$$

for $\eta := 10^{-1}$. The results are shown in Figure 9. As can be seen based on the rightmost column, A is now indeed only approximately of low TT-rank but this does not affect the estimated TT-rank from Algorithm 1.

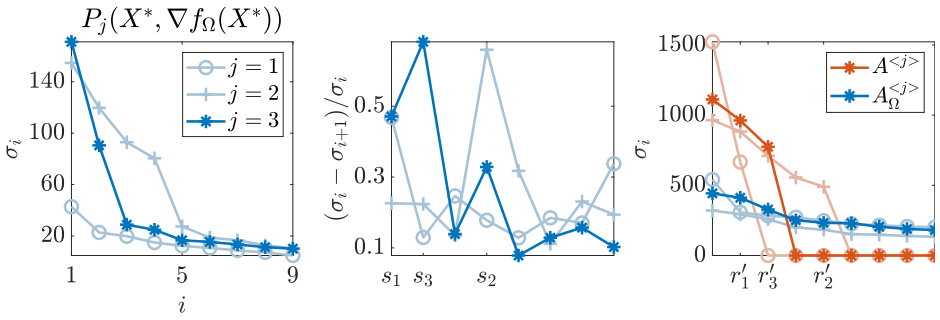

FIGURE 7 *Illustration of the use of Algorithm 1 for asymmetric TT-ranks.* The first 7 singular values of the matrices in (33), the relative gap between these singular values, the singular values of the unfoldings of A_Ω, and A, where X^* is obtained after 15 iterations of the CG algorithm, and with $d := 4$, $n_i := 15$, for $i = 1, \ldots, d$, $r' := [2, 5, 3]$, $r_i := 1$, for $i = 1, \ldots, d-1$, and $\rho_\Omega := 0.3$. The norm of the Riemannian gradient that is obtained at X^* is approximately 13.

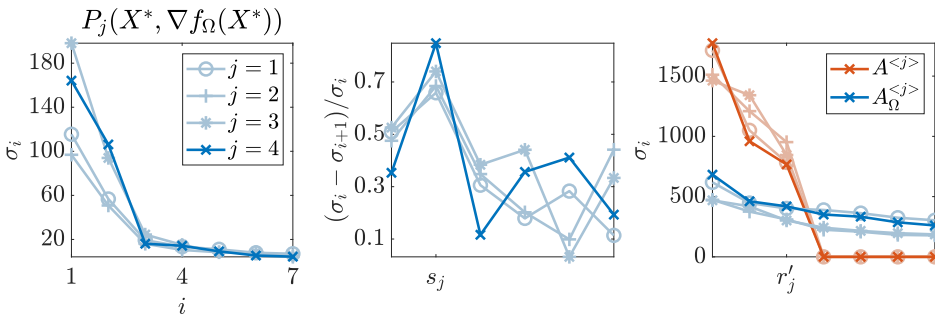

FIGURE 8 *Illustration of the use of Algorithm 1 for order 5.* The first 7 singular values of the matrices in (33), the relative gap between these singular values, the singular values of the unfoldings of A_Ω, and A, where X is obtained after 15 iterations of the CG algorithm, and with $d := 5$, $n_i := 10$, for $i = 1, \ldots, d$, $r'_i := 3$ and $r_i := 1$, for $i = 1, \ldots, d-1$, and $\rho_\Omega := 0.3$. The norm of the Riemannian gradient that is obtained at X^* is approximately 12.

3.1.4 | Algorithm for rank increase

When the search direction $\hat{Y}$ is determined, a line search along this direction is performed. For the LRTCP (1), we can perform an exact line search, i.e., we can compute the step size t_k at iteration k as $t_k = \operatorname{argmin}_{t>0} f_\Omega\left(X^{(k)} + t\hat{Y}\right)$, which can be obtained as

$$\begin{aligned} \frac{\partial f_\Omega\left(X^{(k)} + t\hat{Y}\right)}{\partial t} = \frac{\partial}{\partial t}\left(\frac{1}{2}\left\|X^{(k)}_\Omega + t_k\hat{Y}_\Omega - A_\Omega\right\|^2\right) = 0 \Leftrightarrow \left\langle X^{(k)}_\Omega + t_k\hat{Y}_\Omega - A_\Omega, \hat{Y}_\Omega\right\rangle = 0, \\ \Leftrightarrow t_k = -\frac{\left\langle \nabla f_\Omega\left(X^{(k)}\right), \hat{Y}_\Omega\right\rangle}{\left\|\hat{Y}_\Omega\right\|^2}, \end{aligned} \tag{45}$$

where $\hat{Y}_\Omega$ is defined by (2) and $X^{(k)}$ is the current best approximation at iteration k.

To increase the rank r_i, using the parameters U_i and V_{i+1} that we obtain from the projection in Proposition 1, we need the following retraction operator [18] which projects $X + t\hat{Y}$ to the manifold $\mathbb{R}^{n_1 \times \cdots \times n_d}_{(r_1, \ldots, r_{i-1}, k_i, r_{i+1}, \ldots, r_{d-1})}$:

$$\mathcal{R}\left(X, t, \hat{Y}\right) = X'_{1:(i-1)} \cdot \begin{bmatrix} X'_i & U_i \end{bmatrix} \cdot \begin{bmatrix} X_{i+1} \\ tV_{i+1} \end{bmatrix} \cdot X''_{(i+2):d}, \tag{46}$$

where $\hat{Y} = X'_{1:(i-1)} \cdot U_i \cdot V_{i+1} \cdot X''_{(i+2):d} \in T^\perp_X \mathbb{R}^{n_1 \times \cdots \times n_d}_{\leq (r_{1:(i-1)}, k_i, r_{(i+1):d})}$.

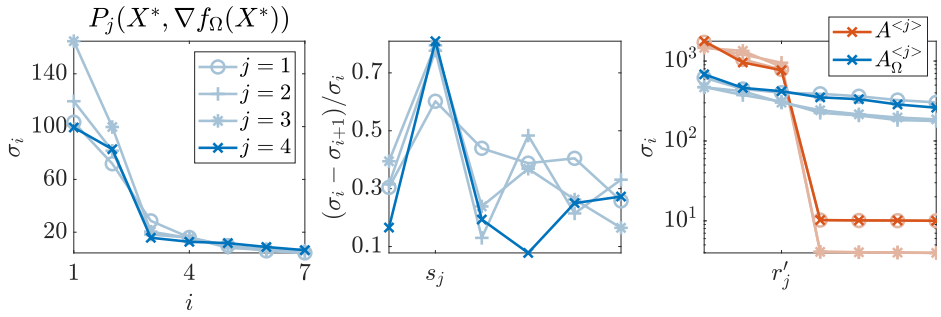

FIGURE 9 *Illustration of the use of Algorithm 1 for tensors of approximate low rank.* The first 7 singular values of the matrices in (33), the relative gap between these singular values, the singular values of the unfoldings of A_Ω, and A, where noise is added to the data as in (44) of size 10^{-1}, and X is obtained after 15 iterations of the CG algorithm, and with $d := 5$, $n_i := 10$, for $i = 1, \ldots, d$, $r'_i := 3$ and $r_i := 1$, for $i = 1, \ldots, d-1$, and $\rho_\Omega := 0.3$. The norm of the Riemannian gradient obtained at X^* is approximately 77.

We do this for $i = 1, \ldots, d-1$. For each i, the rank increase is determined by the estimated rank from Proposition 2. An overview of the algorithm is given in Algorithm 2. Remark that the rank is only increased if the direction produces a sufficient decrease in the cost and test function, f_Ω and f_Γ, respectively. This is controlled by a small parameter ε.

Algorithm 2 Method to increase the rank of a TTD in an RRAM based on projections of the negative gradient onto subcones of the tangent cone.

Input: cost set Ω, test set Γ, $X \in \mathbb{R}^{n_1 \times \cdots \times n_d}_{\leq (r_1, \ldots, r_{d-1})}$, $s_{\max}, r_{\max} \in \mathbb{N}_0^{d-1}$, $0 \leq \varepsilon < 1$

for $i = 1, \ldots, d-1$ **do**
  $P \leftarrow P_i(X, \nabla f_\Omega(X))$ (33);
  $s \leftarrow \min(r_{\max}(i) - r_i, s_{\max}(i))$;
  $s_i \leftarrow \tilde{r}_s(P)$ (41);
  $[U_i^{\mathrm{R}}, S, V] \leftarrow \mathrm{SVD}_s(P)$;
  $V_{i+1}^{\mathrm{L}} \leftarrow SV^\top$;
  $\hat{Y} \leftarrow X = X'_{1:(i-1)} \cdot X'_i \cdot \dot{X}_{i+1} \cdot X''_{(i+2):d}$;
  $\hat{Y}_i \leftarrow U_i$;
  $\hat{Y}_{i+1} \leftarrow V_{i+1}$;
  $t \leftarrow -\frac{\langle \hat{Y}_\Omega, \nabla f_\Omega(X) \rangle}{\|\hat{Y}_\Omega\|^2}$;
  $X_{\mathrm{new}} \leftarrow \mathcal{R}(X, t, \hat{Y})$ (46);
  if $(f_\Omega(X) - f_\Omega(X_{\mathrm{new}})) > \varepsilon$ & $(f_\Gamma(X) - f_\Gamma(X_{\mathrm{new}})) > \varepsilon$ **then**
    $X \leftarrow X_{\mathrm{new}}$;
  end if
end for

Output: X

3.2 | Rank reduction

To reduce the rank, we propose to use the TT-rounding algorithm [22, Algorithm 2], shown in Algorithm 3. The rank reduction is performed from left to right. Remark that we could also perform the reduction from right to left and compare the result but

this would require double the computational effort and is thus not implemented. In the next theorem, we show that Algorithm 3 can be considered as an approximate projection and satisfies a certain angle condition.

Algorithm 3 Rank reduction [22, Algorithm 2]

Input: $A = \dot{A}_1 \cdot A_2'' \cdots A_d'' \in \mathbb{R}^{n_1 \times \cdots \times n_d}_{\leq (r_1', \ldots, r_{d-1}')}, r_i \in \mathbb{N}_0 \ \ \forall i \in \{1, \ldots, d-1\}$

1: **for** $i = 1, \ldots, d-1$ **do**

2: $\quad [X_i'^{\mathrm{R}}, S_i, V_i] \leftarrow \mathrm{SVD}_{r_i}\left(\dot{A}_i^{\mathrm{R}}\right)$;

3: $\quad \dot{A}_{i+1} \leftarrow S_i V_i^\top \cdot A_{i+1}''$;

4: **end for**

5: $\dot{X}_d \leftarrow \dot{A}_d$;

Output: $X = X_1' \cdots X_{d-1}' \cdot \dot{X}_d \in \mathbb{R}^{n_1 \times \cdots \times n_d}_{\leq (r_1, \ldots, r_{d-1})}$

Theorem 2 (Angle condition Algorithm 3)**.** *Let $A \in \mathbb{R}^{n_1 \times \cdots \times n_d}_{\leq (r_1', \ldots, r_{d-1}')}$. Algorithm 3 can be considered as an approximate projection:*

$$\tilde{\mathcal{P}}_{\mathbb{R}^{n_1 \times \cdots \times n_d}_{\leq (r_1, \ldots, r_{d-1})}} : \mathbb{R}^{n_1 \times \cdots \times n_d}_{\leq (r_1', \ldots, r_{d-1}')} \multimap \mathbb{R}^{n_1 \times \cdots \times n_d}_{\leq (r_1, \ldots, r_{d-1})},$$

that satisfies the necessary condition (28) and the angle condition (31) with

$$\omega = \sqrt{\frac{\prod_{i=2}^{d-1} r_i}{\prod_{i=2}^{d-1} r_i'}}. \tag{47}$$

Proof. We first prove that every output X of Algorithm 3 satisfies the necessary condition (28). In the first iteration, it holds that $X_1' = \dot{A}_1 V_1 S_1^{-1}$ and in every subsequent iteration $i = 2, \ldots, d-1$ it holds that $X_i' = S_{i-1} V_{i-1}^\top \cdot A_i'' \cdot V_i S_i^{-1}$, and in the last iteration $\dot{X}_d = S_{d-1} V_{d-1} A_d''$. Thus, the X that is given to the output can be written as

$$X = X_1' \cdots X_{d-1}' \cdot \dot{X}_d = \dot{A}_1 P_{V_1} \cdot A_2'' \cdots P_{V_{d-1}} A_d''.$$

Furthermore, every tensor A can be written as

$$A = \dot{A}_1 \left(P_{V_1} + P_{V_1}^\perp\right) \cdot A_2'' \cdots \left(P_{V_{d-1}} + P_{V_{d-1}}^\perp\right) \cdot A_d'',$$

which can be expanded into a sum of 2^{d-1} terms. For every term, except for the one that equals X, it holds that there exists an index $i \in \{1, \ldots, d-1\}$ such that the first i factors are

$$\dot{A}_1 P_{V_1} \cdot A_2'' \cdots P_{V_{i-1}} \cdot A_i'' \cdot P_{V_i}^\perp = X_1' \cdots X_{i-1}' \cdot S_{i-1} V_{i-1}^\top \cdot A_i'' \cdot P_{V_i}^\perp = X_1' \cdots X_{i-1}' \cdot \dot{A}_i \cdot P_{V_i}^\perp.$$

Thus, it holds that

$$\begin{aligned}\left(X_1' \cdots X_i'\right)^{\mathrm{R},\top} \left(X_1' \cdots X_{i-1}' \cdot \dot{A}_i \cdot P_{V_i}^\perp\right)^{\mathrm{R}} &= X_i'^{\mathrm{R},\top} \left(I_{n_i} \otimes \left(X_1' \cdots X_{i-1}'\right)^{\mathrm{R},\top}\right) \left(I_{n_i} \otimes \left(X_1' \cdots X_{i-1}'\right)^{\mathrm{R}}\right) \left(\dot{A}_i \cdot P_{V_i}^\perp\right)^{\mathrm{R}} \\ &= X_i'^{\mathrm{R},\top} \dot{A}_i^{\mathrm{R}} P_{V_i}^\perp = S_i V_i^\top P_{V_i}^\perp = 0,\end{aligned}$$

and because of (17), the inner product of A with X is $\langle X, A\rangle = \langle X, X\rangle$, and the angle condition can be written in the form (31).

Let $\hat{X} = \hat{X}_1' \cdots \hat{X}_{d-1}' \cdot \hat{X}_d \in \mathcal{P}_{\mathbb{R}^{n_1 \times \cdots \times n_d}_{\leq (r_1, \ldots, r_{d-1})}} A$, then $\mathcal{P}_{T_{\hat{X}} \mathbb{R}^{n_1 \times \cdots \times n_d}_{(r_1, \ldots, r_{d-1})}} \nabla f\left(\hat{X}\right) = 0$ and $\hat{X}$ can be written as in Theorem 1:

$$\hat{X} = \dot{A}_1 \cdot D_1 \cdot A_2'' \cdots D_{d-1} \cdot A_d'',$$

where based on (35) and (36): $\hat{X}'_1 = \dot{A}_1 C_1 Q_1^{-1}$, $\hat{X}'_i = B_i^\top Q'_{i-1} \cdot A''_i \cdot C_i Q_i^{-1}$, for $i = 2, \dots, d-1$, and $\hat{X}_d = B_d^\top Q'_{d-1} A''_d$. Furthermore, $\hat{X}$ must by definition also satisfy the necessary condition $\langle \hat{X}, A \rangle = \|\hat{X}\|^2$. Additionally, $\|\hat{X}\| \leq \|A\|$, and more specifically

$$\begin{aligned} \|\hat{X}\| = \|\dot{A}_1 \cdot D_1 \cdot A''_2 \cdots D_{d-1} \cdot A''_d\| &\leq \|\dot{A}_1 \cdot D_1 \cdot A''_2 \cdots D_{d-2} \cdot A''_{d-1} \cdot A''_d\| \\ &\vdots \\ &\leq \|\dot{A}_1 \cdot D_1 \cdot A''_2 \cdots A''_d\|. \end{aligned} \tag{48}$$

These inequalities hold because

$$\begin{aligned} \|\dot{A}_1 D_1 \cdot A''_2 \cdots D_i \cdot A''_{i+1} \cdots A''_d\| &= \|\dot{A}_1 D_1 \cdot A''_2 \cdot D_2 \cdots A''_i \cdot D_i\| \\ &= \|\dot{A}_1 C_1 Q_1^{-1} B_2^\top Q'_1 \cdot A''_2 \cdot C_2 Q_2^{-1} B_3^\top Q'_2 \cdots A''_i \cdot C_i Q_i^{-1} B_{i+1}^\top Q'_i\| \\ &= \|\hat{X}'_1 \cdots \hat{X}'_i \cdot B_{i+1}^\top Q'_i\| \\ &= \|B_{i+1}^\top Q'_i\|, \end{aligned}$$

and from (40) we know that $B_{i+1}^\top = X_i'^{\mathrm{R},\top} \left(B_i^\top \cdot A'_i\right)^{\mathrm{R}}$. Thus,

$$\|B_{i+1}^\top Q'_i\| = \left\|X_i'^{\mathrm{R},\top} \left(B_i^\top \cdot A'_i\right)^{\mathrm{R}} Q'_i\right\| = \left\|X_i'^{\mathrm{R},\top} \left(B_i^\top Q'_{i-1} \cdot A''_i\right)^{\mathrm{R}}\right\| = \left\|P_{X_i'^{\mathrm{R}}} \left(B_i^\top Q'_{i-1} \cdot A''_i\right)^{\mathrm{R}}\right\| \leq \left\|B_i^\top Q'_{i-1} \cdot A''^{\mathrm{L}}_i\right\| = \|B_i^\top Q'_{i-1}\|,$$

and consequently (48) holds.

We can now prove the angle condition (31) with ω as in (47). In the first iteration, $X_1'^{\mathrm{R}}$ is obtained from the truncated SVD of $\dot{A}_1$ and thus it holds that

$$\begin{aligned} \|\dot{A}_1 P_{V_1}\| = \|\dot{A}_1 P_{V_1} \cdot A''_2 \cdots A''_d\| = \|P_{X'_1} A^{\mathrm{L}}\| &\overset{(A1)}{\geq} \|P_{\hat{X}'_1} A^{\mathrm{L}}\| = \left\|\dot{A}_1 C_1 Q_1^{-1} \left(C_1 Q_1^{-1}\right)^\top Q_1'^\top A_1'^\top A'_1 \cdot \dot{A}_2\right\| \\ &= \|\dot{A}_1 C_1 Q_1^{-1} B_2^\top Q'_1 \cdot A''_2\| \\ &= \|\dot{A}_1 D_1\| \overset{(48)}{\geq} \|\hat{X}\|, \end{aligned} \tag{49}$$

where we used the fact that $B_2 = Q'_1 C_1 Q_1^{-1}$ as in Theorem 1. In the next iteration, V_2 is obtained from the truncated SVD of $\left(S_1 V_1^\top \cdot A''_2\right)^{\mathrm{R}}$ and thus

$$\begin{aligned} &\left\|\left(S_1 V_1^\top \cdot A''_2\right)^{\mathrm{R}} P_{V_2}\right\|^2 \overset{(A2)}{\geq} \frac{r_2}{r'_2} \left\|S_1 V_1^\top \cdot A''_2\right\|^2 \\ \Leftrightarrow &\left\|X'_1 \left(S_1 V_1^\top \cdot A''_2 \cdot P_{V_2}\right)^{\mathrm{L}}\right\|^2 \geq \frac{r_2}{r'_2} \left\|X'_1 \left(S_1 V_1^\top \cdot A''_2\right)^{\mathrm{L}}\right\|^2 \\ \Leftrightarrow &\left\|\dot{A}_1 P_{V_1} \cdot A''_2 \cdot P_{V_2}\right\|^2 \geq \frac{r_2}{r'_2} \left\|\dot{A}_1 P_{V_1}\right\|^2, \end{aligned}$$

where we used the fact that multiplication on the left or right by an orthogonal matrix does not change the norm, as well as left or right reshaping the tensor. In the third iteration, we can apply this principle again to obtain

$$\left\|\dot{A}_1 P_{V_1} \cdot A''_2 \cdot P_{V_2} \cdot A''_3 \cdot P_{V_3}\right\|^2 \overset{(A2)}{\geq} \frac{r_3}{r'_3} \left\|\dot{A}_1 P_{V_1} \cdot A''_2 \cdot P_{V_2}\right\|^2.$$

We can do this recursively and combine the inequalities to obtain

$$\left\|\dot{A}_1 P_{V_1} \cdot A''_2 \cdot P_{V_2} \cdot A''_3 \cdots P_{V_{d-1}} A''_d\right\|^2 \geq \frac{\prod_{i=2}^{d-1} r_i}{\prod_{i=2}^{d-1} r'_i} \left\|\dot{A}_1 P_{V_1} \cdot A''_2 \cdots A''_d\right\|^2.$$

And thus using (49), we obtain

$$\left\|\dot{A}_1 P_{V_1} \cdot A''_2 \cdot P_{V_2} \cdot A''_3 \cdots P_{V_{d-1}} A''_d\right\|^2 = \|X\|^2 \geq \frac{\prod_{i=2}^{d-1} r_i}{\prod_{i=2}^{d-1} r'_i} \|\hat{X}\|^2,$$

from which the angle condition follows. □

3.2.1 | Numerical rank

To determine how much the rank is decreased, the following *numerical* Δ*-rank* is defined, inspired by [7].

Definition 2 (Numerical matrix Δ-rank)**.** Given $\Delta \in [0,1]$, the Δ*-rank* of $B \in \mathbb{R}^{m\times n}$ is defined as

$$\operatorname{rank}_\Delta B := \begin{cases} 0 & \text{if } B = 0 \\ \min\left\{ j \in \{1,\dots,\operatorname{rank} B\} \mid \frac{\sigma_j(B)-\sigma_{j+1}(B)}{\sigma_j(B)} \geq \Delta \right\} & \text{otherwise} \end{cases},$$

where as in (41) $\sigma_j(B)$, for $j = 1,\dots,\operatorname{rank} B$, denote the singular values of B in decreasing order and $\sigma_{j+1}(B) = 0$ for $j = \operatorname{rank} B$.

For $\Delta := 0$, the Δ-rank equals one, and for $\Delta := 1$, it equals the standard rank. Remark that the standard definition of numerical matrix rank uses an absolute threshold on the singular values as in the following definition.

Definition 3 (Numerical matrix rank)**.** Given $\epsilon \geq 0$, the *numerical rank* of $B \in \mathbb{R}^{m\times n}$ is defined as

$$\operatorname{rank}_\epsilon B := \min\left\{ j \in \{1,\dots,\operatorname{rank} B\} \mid \sigma_{j+1}(B) \leq \epsilon \right\}.$$

The squared norm of the error made by truncating a matrix at this rank, e.g., using a truncated SVD, can then be upper bounded by $(\operatorname{rank} B - j)\epsilon^2$. For the Δ-rank, this error can be expressed as: $(\operatorname{rank} B - j)(1-\Delta)^2\sigma_j(B)^2$. The following definition generalizes the Δ-rank to tensors.

Definition 4 (Numerical tensor Δ-rank)**.** Given $\Delta \in [0,1]$, the Δ*-rank* of $X \in \mathbb{R}^{n_1\times\cdots\times n_d}$ is defined as

$$\operatorname{rank}_\Delta(X) := \left(\operatorname{rank}_\Delta X^{<1>},\ \operatorname{rank}_\Delta X^{<2>},\ \dots,\ \operatorname{rank}_\Delta X^{<d-1>}\right),$$

where the Δ-rank for matrices was defined in Definition 2.

Remark that the computation of the SVD of these unfoldings can be computationally expensive for high-dimensional tensors. However, when X is already given as a TTD, the cost can be reduced significantly by using the terms $\dot{X}_i$. The Δ-rank then becomes:

$$\operatorname{rank}_\Delta(X) = \left(\operatorname{rank}_\Delta \dot{X}_1,\ \operatorname{rank}_\Delta \dot{X}_2^{\mathrm{R}},\ \dots,\ \operatorname{rank}_\Delta \dot{X}_{d-1}^{\mathrm{R}}\right). \tag{50}$$

This is possible because

$$X^{<i>} = \left(I_{n_i} \otimes (X_1' \cdots X_{i-1}')^{\mathrm{R}}\right) \dot{X}_i^{\mathrm{R}} \left(X_{i+1}'' \cdots X_d''\right)^{\mathrm{L}},$$

and thus the singular values of $X^{<i>}$ and $\dot{X}_i^{\mathrm{R}}$ are the same.

To compute the error that is made if we would use the TT-SVD algorithm, see Algorithm 3, to truncate X at this rank, we can use [22, Theorem 2.2], which results in the following proposition.

Proposition 3 (Error TT-SVD, Algorithm 3)**.** *If Algorithm 3 is used to compute a truncated TTD X of A at $r := (r_1,\cdots,r_{d-1}) = \operatorname{rank}_\Delta A$ and if $\operatorname{rank}_{\mathrm{TT}}(A) := (r_1',\cdots,r_{d-1}')$, then*

$$\|X-A\|^2 \leq \sum_{i=1}^{d-1} (r_i' - r_i)(1-\Delta)^2 \sigma_{r_i}(\dot{A}_i)^2. \tag{51}$$

Proof. From [22, Theorem 2.2], it is known that if

$$A^{<i>} = R_i + E_i, \tag{52}$$

where $\operatorname{rank} R_i = r_i$ and $\|E_i\|^2 = \varepsilon_i^2$, then

$$\|X-A\|^2 \leq \sum_{i=1}^{d-1} \varepsilon_i^2.$$

In our case, we know that the truncated SVD can approximate $A^{<i>}$ with an error bounded by $(r_i' - r_i)(1-\Delta)^2\sigma_{r_i}(\dot{A}_i)^2$. Consequently, (51) directly follows. □

In the experiments in 4.4, Δ is set to 0.8.

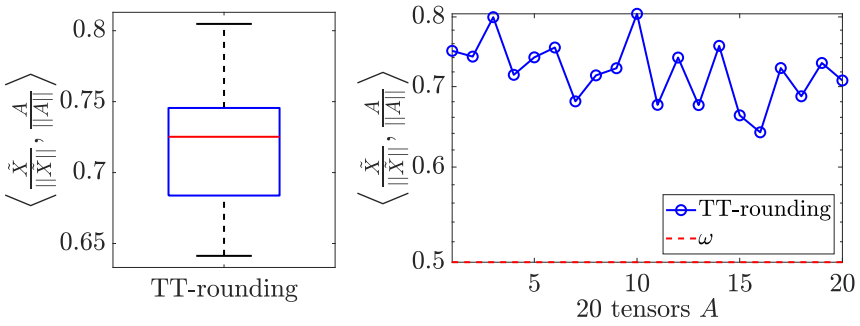

FIGURE 10 *The derived angle condition is a pessimistic lower bound for randomly generated tensors.* Illustration of the angle condition of Algorithm 3 for 20 randomly generated tensors A as in (43) for $d := 4$, $n_i := 10$, for $i = 1, \ldots, 4$, $r'_i := 4$ and $r_i := 2$, for $i = 1, \ldots, 3$.

3.2.2 | Experiment rank reduction

An illustration of the angle condition in Theorem 2 is shown in Figure 10. For this experiment, A is generated as in (43) with $d := 4$, $n_i := 10$, for $i = 1, \ldots, 4$, and $r'_i := 4$, for $i = 1, \ldots, 3$. The rank of the lower-rank set was chosen as $r_i := 2$, for $i = 1, \ldots, 3$. We compute the value

$$\left\langle \frac{\tilde{X}}{\|\tilde{X}\|}, \frac{A}{\|A\|} \right\rangle, \tag{53}$$

for the different random tensors A and where $\tilde{X}$ is obtained with Algorithm 3 because the value $\|\mathcal{P}_{\mathbb{R}^{n_1 \times \cdots \times n_d}_{(r_1, \ldots, r_{d-1})}} A\|$ is not known. Remark that

$$\left\langle \frac{\tilde{X}}{\|\tilde{X}\|}, \frac{A}{\|\mathcal{P}_{\mathbb{R}^{n_1 \times \cdots \times n_d}_{(r_1, \ldots, r_{d-1})}} A\|} \right\rangle \geq \left\langle \frac{\tilde{X}}{\|\tilde{X}\|}, \frac{A}{\|A\|} \right\rangle.$$

Thus, if (53) is larger than ω, also the angle condition is satisfied. In the left subfigure in Figure 10, the box plot for this experiment is shown and on the right, the value of (53) for each tensor A and corresponding image of the approximate projection $\tilde{X}$. As can be seen, the value in (53) is always larger than ω, which equals 0.5 and proves that the angle condition is satisfied for this experiment.

4 | RIEMANNIAN RANK-ADAPTIVE METHOD

Now that all required elements are explained, the overall layout of the RRAM can be given and is shown in Algorithm 4. The layout is inspired by the state-of-the-art methods for matrix and tensor completion [7, 36, 26]. The main contributions are the method to increase the rank using Algorithm 2 on line 20 and the method to decrease the rank on line 14 with Algorithm 3.

The optimization on the smooth, fixed-rank manifold is displayed on line 3. We use a Riemannian conjugate gradient (CG) algorithm developed for tensor completion [26] for this purpose. This choice also allows for a good comparison with the state-of-the-art RRAM [26] in terms of rank adaptation. The RRAM and CG algorithm are available in the Manopt toolbox [1].

4.1 | Computational complexity

The CG algorithm has a complexity of $\mathcal{O}\left(dnr^3 + d|\Omega|r^2\right)$ per iteration. Since the number of samples must be larger than or equal to the dimension of the manifold [2], the complexity per iteration is at least $\mathcal{O}\left(d^2nr^4\right)$. An efficient implementation of the TT-rounding algorithm to reduce the rank requires $\mathcal{O}\left(dnr^2 + dr^4\right)$ operations [22]. The computation of the matrices $P_i(X, \nabla f(X))$ requires approximately the same computational effort as the computation of the Riemannian gradient in the CG algorithm [27], which is $\mathcal{O}\left(dnr^3 + d|\Omega|r^2\right)$ or $\mathcal{O}\left(d^2nr^4\right)$ if $|\Omega| = \mathcal{O}\left(dnr^2\right)$. The computation of the SVD_{s_i}, where $s_i \approx r$, of the matrices $P_i(X, \nabla f(X))$ is $\mathcal{O}\left(r^3n^2\right)$. This is a complicated step in the algorithm if $|\Omega| < rn(n-d)/d$, which is not the case for the experiments in Section 4.4. These two steps have to be computed d times in Algorithm 2.

Algorithm 4 RRAM$^d_{\tilde{\mathcal{P}}}$ for higher-order tensor completion in the TT format.

Input: $\Omega, \Gamma, A_\Omega, A_\Gamma, X^{(0)}, \varepsilon_\Omega, \varepsilon_R \in (0,1), \varepsilon_\Gamma \geq 0, \Delta \in [0,1), j_{\max}, k_{\max} \in \mathbb{N}_0, r_{\max}, s_{\max} \in \mathbb{N}_0^{d-1}, \varepsilon > 0$

1: $X_{\text{new}} \leftarrow X^{(0)}$;
2: **for** $k = 1, \ldots, k_{\max}$ **do**
3: $X^{(k)} \leftarrow$ `Riem_conjugate_gradient` $(X_{\text{new}}, \varepsilon_\Omega, \varepsilon_R, j_{\max})$; [26]
4: **if** $\rho_\Gamma\left(X^{(k)}\right)$ (see (54)) $> \varepsilon_\Gamma$ **then**
5: $X^{(k)} \leftarrow X^{(k-1)}$;
6: go to output;
7: **else if** $\frac{\sqrt{2 f_\Omega(X^{(k)})}}{\|A_\Omega\|} < \varepsilon_\Omega$ **then**
8: go to output;
9: **end if**
10: **if** $k < k_{\max}$ **then**
11: $r \leftarrow \operatorname{rank} X^{(k)}$;
12: $r_\Delta \leftarrow \operatorname{rank} X_\Delta$ (50);
13: **if** $\sum_{i=1}^{d-1} r_\Delta(i) < \sum_{i=1}^{d-1} r(i)$ **then**
14: $X_\Delta \leftarrow$ Algorithm 3(X, r_Δ);
15: **if** $f_\Omega(X_\Delta) < f_\Omega\left(X^{(k)}\right)$ **or** $f_\Gamma(X_\Delta) < f_\Gamma\left(X^{(k)}\right)$ **then**
16: $X_{\text{new}} \leftarrow X_\Delta$;
17: go to line 2 and continue for loop;
18: **end if**
19: **end if**
20: $X_{\text{new}} \leftarrow$ Algorithm 2$\left(\Omega, \Gamma, X^{(k)}, r_{\max}, s_{\max}, \varepsilon\right)$;
21: **end if**
22: **end for**

Output: X_k

4.2 | Overfitting

As already briefly discussed in the introduction, overfitting is a common difficulty for the tensor completion problem (1). Overfitting occurs if k is too high in at least one mode. In this case, X can still approximate A_Ω well (small $f_\Omega(X)$), but $f(X)$ is large, assuming A is known. That is why usually in the literature a test set Γ is included [7, 4, 32, 14, 26]. The corresponding error function is $f_\Gamma(X)$. If

$$\rho_\Gamma\left(X^{(k)}\right) := \frac{f_\Gamma\left(X^{(k)}\right) - f_\Gamma\left(X^{(k-1)}\right)}{f_\Gamma\left(X^{(k-1)}\right)} \tag{54}$$

is larger than some positive constant ε_Γ at iteration k, the RRAM is stopped and $X^{(k-1)}$ is considered as the best solution. The set Γ is typically smaller than Ω, e.g., $|\Gamma| = 0.25 \cdot |\Omega|$.

4.3 | Stopping criterion

The inner Riemannian CG algorithm runs until one of the following conditions is satisfied:

- a maximal number of inner iterations $j_{\max}$ is reached,
- the norm of the Riemannian gradient is smaller than ε_R,

- $\sqrt{2f_\Omega(X^{(j)})}/\|A_\Omega\|$ is smaller than ε_Ω,
- $\frac{\sqrt{2f_\Omega(X^{(j)})}-\sqrt{2f_\Omega(X^{(j-1)})}}{\sqrt{2f_\Omega(X^{(j)})}}$ is smaller than a certain tolerance, which is by default set to 10^{-8}. We did not change this value.

On the other hand, Algorithm 4 runs until

- the maximal number of outer iterations $k_{\max}$ is reached,
- $\sqrt{2f_\Omega(X^{(k)})}/\|A_\Omega\|$ is smaller than ε_Ω,
- the relative error on the test set $\rho_\Gamma(X^{(k)})$ in (54) has increased more than ε_Γ.

A convergence analysis is out of the scope of this paper but might be interesting to perform in future work.

In the following subsection, several numerical experiments with Algorithm 4 are given. Additionally, one experiment is provided to detangle the contribution of the projection and the rank estimation.

4.4 | Experiments RRAM

In this section, we compare the results that we obtain with the algorithm proposed in Algorithm 4 with the one from [26]. Both algorithms are denoted by $\text{RRAM}^d_{\tilde{\mathcal{P}}}$ and $\text{RRAM}_{\text{randn}}$ respectively. In the first subsection A is generated as in (43) of known low TT-rank and in the second subsection, A is obtained by evaluating an exponential function in four variables in equidistant points.

4.4.1 | Synthetic data

In this subsection, we generate A and X_0 as in (43) for different values of d, n, and r'. The following input parameters for Algorithm 4 were chosen:

$$\begin{aligned} &\varepsilon_\Omega := 10^{-8}, && \varepsilon := 10^{-10}, && j_{\max} := 15, && s_{\max}(i) := 8, && \\ &\varepsilon_{\text{R}} := 10^{-8}, && \varepsilon_\Gamma := 1, && k_{\max} := 15, && r_{\max}(i) := 10, && \Delta := 0.8, \end{aligned}$$

for $i = 1, \ldots, d-1$. Furthermore, $\text{RRAM}_{\text{randn}}$ needs as input an upper bound on the TT-rank. However, there is no possibility to use different values for each mode. Thus, we gave in all experiments in this section $\max_i(r'_i)$ as input. This is in practice of course not possible when the rank is not known.

In the first experiment, we choose $d := 5$, $n_i := 20$, for $i = 1, \ldots, 4$, $r'_i := 4$, for $i = 1, \ldots, 3$, and $\rho_\Omega := 0.1$. The samples are generated in Matlab using `randperm` over all indices. The results are shown in Figure 11. In the left subfigure, the convergence of the relative value of the cost function (w.r.t. the norm of A_Ω) is shown for both algorithms. The x-axis indicates the cumulative number of inner iterations of the CG algorithm. The middle subfigure shows the convergence of the relative value of the test function. Now the x-axis shows the time in seconds. The rightmost column of subfigures shows the evolution of the TT-rank. As explained in the introduction, the rank increase of $\text{RRAM}_{\text{randn}}$ is fixed and increases always by one. This algorithm thus needs 13 outer iterations, 200 inner iterations, or approximately 100 seconds to reach the rank of A and obtain a numerically exact approximation. On the other hand, the proposed RRAM can detect the TT-rank of A after the first outer iteration and converges to a numerically exact approximation in approximately 10 seconds and 25 inner iterations.

In a second experiment, noise is added to the tensor from the previous experiment as in (44) of size 10^{-1}. The results are shown in Figure 12. Because of the noise, the most accurate value of the relative cost function obtained is approximately $2 \cdot 10^{-5}$. After the first outer iteration, the estimated TT-rank is slightly too high but after the second outer iteration, the rank is reduced to the correct value. The tolerance on the cost function ε_Ω was lowered to 10^{-2}. This value is however not reached and Algorithm 4 stops because the tolerance on the gradient, equal to 10^{-8}, is reached. Still $\text{RRAM}^d_{\tilde{\mathcal{P}}}$ obtains a solution of the same accuracy faster than $\text{RRAM}_{\text{randn}}$, both in number of iterations as computation time.

In a third example, the order is increased to $d := 6$ and the TT-rank of A is chosen as $r := [2, 4, 5, 4, 2]$. The size of A in each dimension is lowered to 15. The results are shown in Figure 13. Again, the proposed algorithm is significantly faster than the

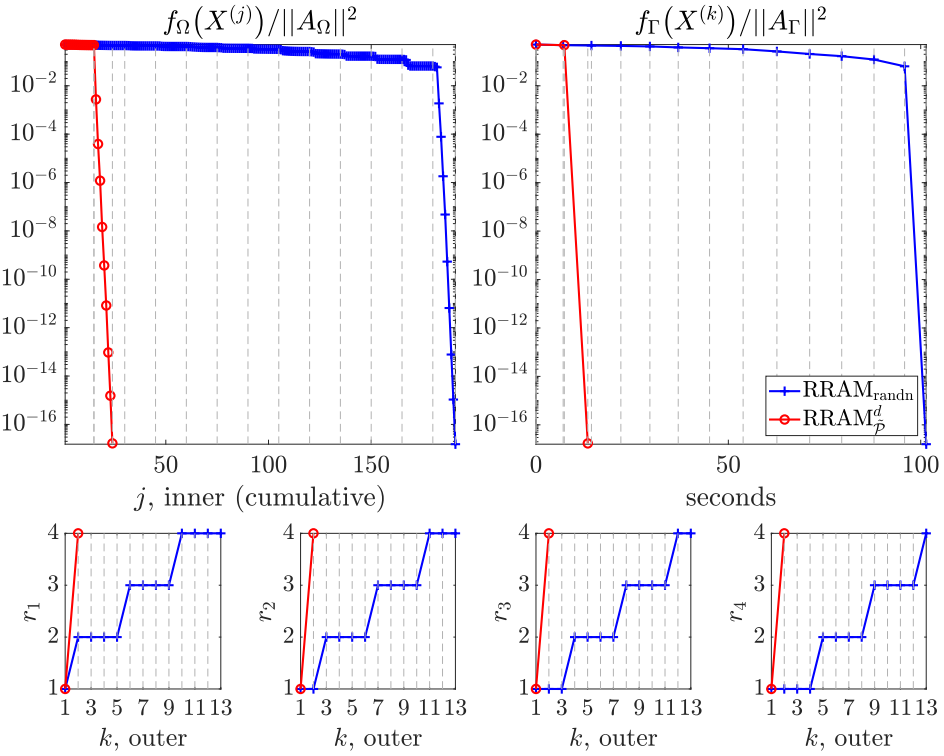

FIGURE 11 *Comparison of the proposed versus the state-of-the-art RRAM for a synthetic TTD A of order 5 and TT-rank 4 in each mode.* Comparison of RRAM$^d_{\tilde{\mathcal{P}}}$ proposed in Algorithm 4 and RRAM$_{\text{randn}}$ [26] for $d := 5$, $n_i := 20$, for $i = 1, \ldots, 5$, $\rho_\Omega := 0.1$, $r'_i := 4$, for $i = 1, \ldots, 4$, and A generated as in (43).

one from [26], both in number of iterations and computation time. Furthermore, the rank is estimated correctly again after only one outer iteration.

The last experiment detangles the contribution of the rank estimation and the projection onto the tangent cone. For this purpose, the setup of the previous experiment is used. The maximal number of iterations is decreased to 10. The projection is compared with subsequent random increases of the TT-rank in each mode as shown in Algorithm 5. Remark that the rank estimation method is still used in this algorithm. The results are shown in Figure 14. As can be seen, because of the projection, the decrease in cost function is faster in the beginning compared to the random directions. However, because of the increased computation time of the projection, this decrease in iterations results in approximately the same total computation time. Consequently, the rank estimation method can be considered the most crucial improvement of Algorithm 4. The projection onto the tangent cone of the gradient can be considered as a warm start strategy of the Riemannian CG algorithm after rank increase as in [31]. Consequently, when computing (a subset of) the singular vectors of the matrices $P_i\left(X, \nabla f(X)\right)$ is costly, e.g., when n_i is large, it can be an option to use random directions in combination with the rank estimation method. Otherwise, we see no reason to use random directions compared to the projection of the gradient. Furthermore, remark that the (first few) singular values of $P_i\left(X, \nabla f(X)\right)$ already have to be calculated to determine the estimated rank.

4.4.2 | Function interpolation

In this section, we reproduce one of the experiments in [26]. The following real function in four variables $x_1, x_2, x_3, x_4 \in [0, 1]$ is considered:

$$f(x_1, x_2, x_3, x_4) := \exp\left(-\left\|x_1 \;\; x_2 \;\; x_3 \;\; x_4\right\|\right).$$

The function is sampled in 20 uniformly distributed points in each variable interval to obtain the following tensor:

$$A(i_1, i_2, i_3, i_4) := \exp\left(-\left\|\tfrac{i_1-1}{19} \;\; \tfrac{i_2-1}{19} \;\; \tfrac{i_3-1}{19} \;\; \tfrac{i_4-1}{19}\right\|\right), \tag{55}$$

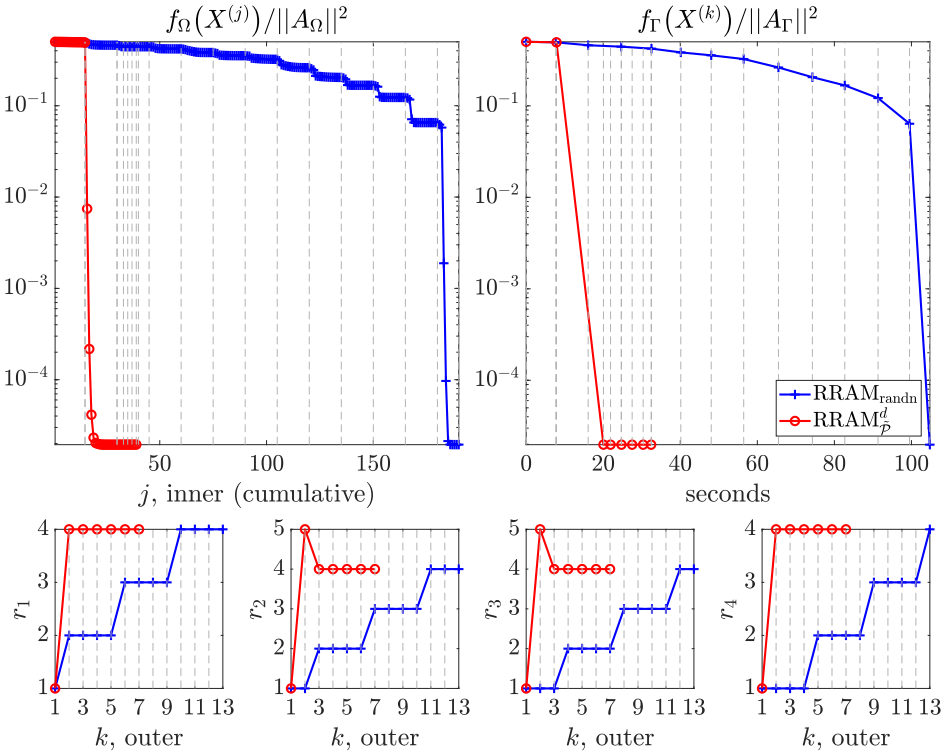

FIGURE 12 *The same experiment as in Figure 11 with noise added of size* 10^{-1}. Comparison of RRAM$^d_{\tilde{\mathcal{P}}}$ proposed in Algorithm 4 and RRAM$_{\text{randn}}$ [26] for $d := 5$, $n_i := 20$, for $i = 1, \ldots, 5$, $\rho_\Omega := 0.1$, $r'_i := 4$, for $i = 1, \ldots, 4$, and A generated as in (43) with noise added of size 10^{-1} as in (44).

Algorithm 5 Method to increase the rank of a TTD in an RRAM using the rank estimation method based on a random direction.

Input: cost set Ω, $X \in \mathbb{R}^{n_1 \times \cdots \times n_d}_{\leq (r_1, \ldots, r_{d-1})}$, $s_{\max}, r_{\max} \in \mathbb{N}_0^{d-1}$, $0 \leq \varepsilon < 1$

1: **for** $i = 1, \ldots, d-1$ **do**

2: $\quad P \leftarrow P_i(X, \nabla f_\Omega(X))$ (33);

3: $\quad s \leftarrow \min(r_{\max}(i) - r_i, s_{\max}(i))$;

4: $\quad s_i \leftarrow \tilde{r}_s(P)$ (41);

5: $\quad X_i^{\mathrm{R}} \leftarrow \left[X_i^{\mathrm{R}},\ 10^{-8} \cdot \mathtt{randn}(r_{i-1} n_i \times s_i)\right]$;

6: $\quad X_{i+1}^{\mathrm{L}} \leftarrow \begin{bmatrix} X_{i+1}^{\mathrm{L}} \\ 10^{-8} \cdot \mathtt{randn}(s_i \times n_{i+1} r_{i+1}) \end{bmatrix}$;

7: **end for**

Output: X

for $i_1, i_2, i_3, i_4 = 1, \ldots, 20$. The input parameters for Algorithm 4 were the same as in the previous subsection, except for $s_{\max}(i) := 5$ and $r_{\max}(i) := 5$, for $i = 1, \ldots, d-1$. The maximal rank for RRAM$_{\text{randn}}$ was also set to 5 in each mode and $\rho_\Omega := 0.1$, both as in the experiment in [26]. The results are shown in Figure 15. As can be seen, Algorithm 4 estimates the TT-rank after each outer iteration as 1 in each mode because the tensor samples an exponential function and consequently also the singular values of the unfoldings of A are decaying exponentially as can be seen in Figure 16. The most accurate relative cost function that both methods are able to obtain is approximately 10^{-9}. This is as expected as A is only approximately of low rank. However, RRAM$^d_{\tilde{\mathcal{P}}}$ is still able to reach the same accuracy in cost and test function faster both in terms of iterations as computation time.

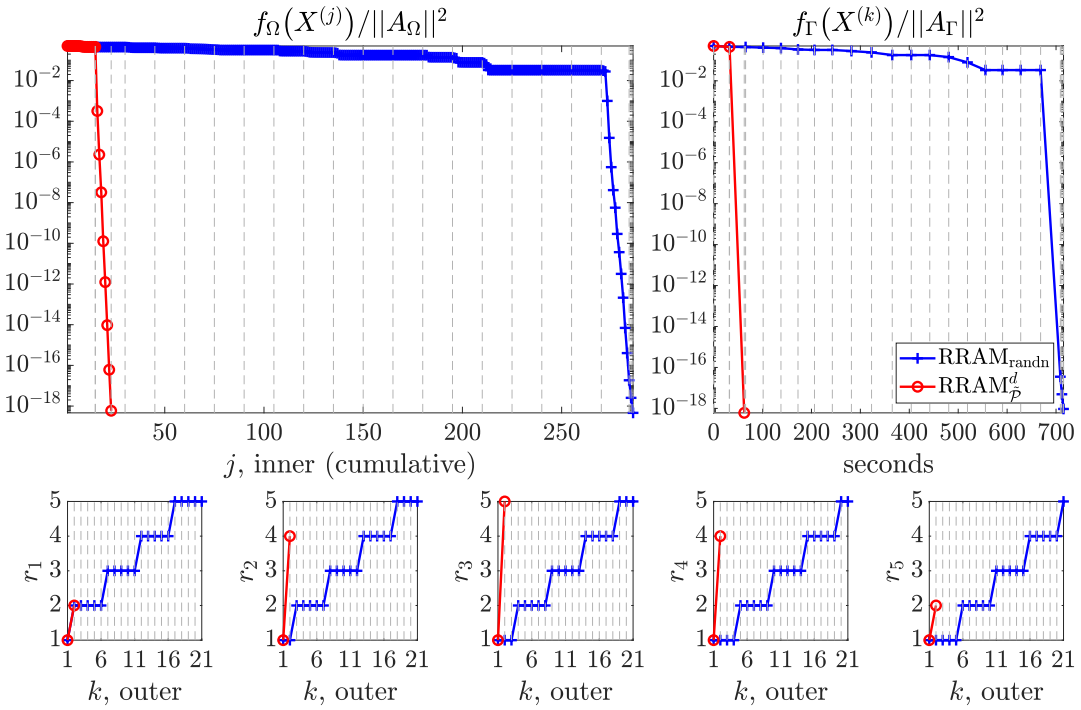

FIGURE 13 *Comparison of the proposed versus the state-of-the-art RRAM for a synthetic TTD A of order 6 and asymmetric TT-ranks.* Comparison of RRAM$^d_{\tilde{\mathcal{P}}}$ proposed in Algorithm 4 and RRAM$_{\text{randn}}$ [26] for $d := 6$, $n_i := 15$, for $i = 1, \ldots, 6$, $\rho_\Omega := 0.1$, $r' := [2, 4, 5, 4, 2]$, and A generated as in (43).

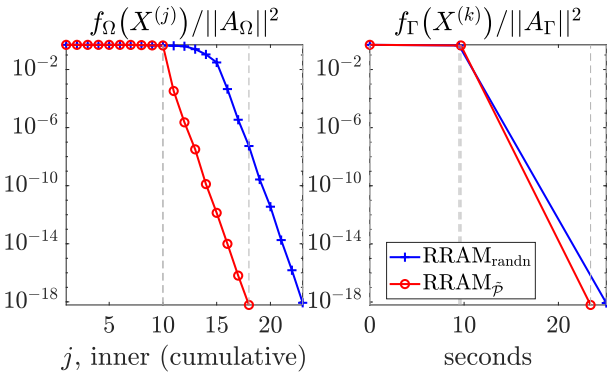

FIGURE 14 *Same experiment as in Figure 13 but with the projection in Algorithm 2 replaced by a random rank increase as shown in Algorithm 5.* Comparison of RRAM$^d_{\tilde{\mathcal{P}}}$ proposed in Algorithm 4 with projection onto the tangent cone as proposed in Algorithm 2, and a random rank increase as in Algorithm 5, for $d := 6$, $n_i := 15$, for $i = 1, \ldots, 6$, $\rho_\Omega := 0.1$, $r' := [2, 4, 5, 4, 2]$, and A generated as in (43).

5 | CONCLUSIONS

A Riemannian rank-adaptive method is proposed for the tensor completion problem in the low-rank tensor-train format which improves the state-of-the-art RRAM by including a method to increase the rank based on successive projections of the negative gradient onto subcones in the normal part of the tangent cone to the variety of bounded tensor-train rank tensors. Furthermore, a rank estimation method is included to estimate a good value for the amount of rank increase. When the tensor to complete is exactly of low rank, the method is able to retrieve this rank. Additionally, when the algorithm converges to an element of a lower-rank set, the rank is reduced based on the TT-rounding algorithm [22], which can be considered as an approximate projection

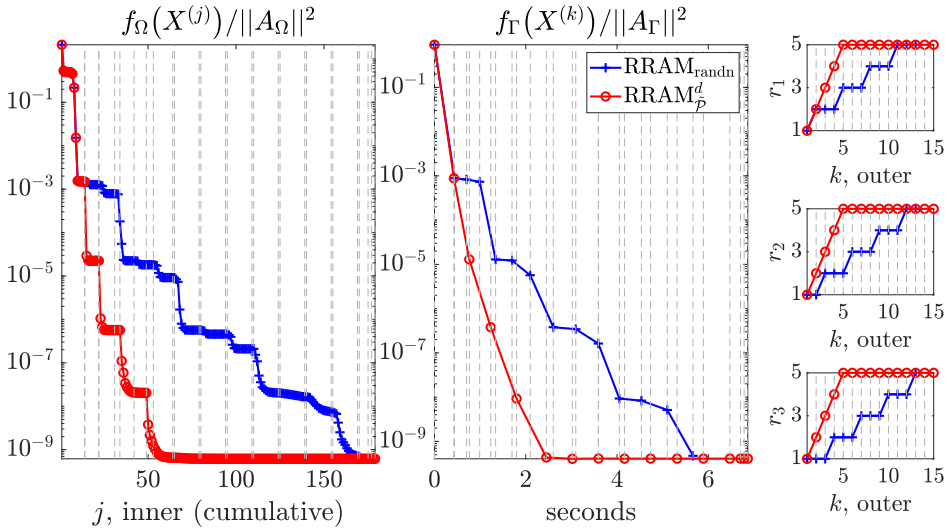

FIGURE 15 *Comparison of the proposed versus the state-of-the-art RRAM for the interpolation of an exponential function in four variables.* Comparison of RRAM$^d_{\mathcal{P}}$ proposed in Algorithm 4 and RRAM$_{\text{randn}}$ [26] for $d := 4$, $n_i := 20$, for $i = 1, \ldots, 4$, $\rho_\Omega := 0.1$, and A generated as in (55).

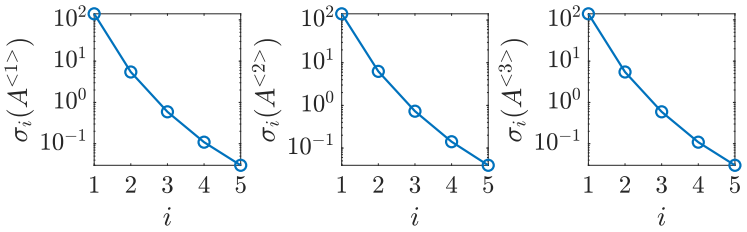

FIGURE 16 *The singular values of the unfoldings of the fourth order tensor obtained when discretizing an exponential function are exponentially decaying.* The first 5 singular values of the unfoldings of A generated as in (55).

onto the lower-rank set and is proven to satisfy a certain angle condition to ensure that the image is sufficiently close to one of the true projection. Several numerical experiments with synthetic tensors and a tensor obtained from function evaluations were given. In all experiments, the proposed RRAM is able to recover the low-rank tensor or obtain a good approximation of the full-rank tensor faster than the state-of-the-art RRAM, both in terms of iterations and computation time.

AUTHOR CONTRIBUTIONS

Charlotte Vermeylen wrote the manuscript and Marc Van Barel reviewed the text and supervised the work.

ACKNOWLEDGMENTS

This work was funded by (1) the Flemish Government: this research received funding under the AI Research Program. Charlotte Vermeylen is affiliated to Leuven.AI - KU Leuven institute for AI, B-3000, Leuven, Belgium. (2) KU Leuven Internal Funds: iBOF/23/064, C14/22/096 and IDN/19/014. Marc Van Barel was supported by the Research Council KU Leuven, C1-project C14/17/073 and by the Fund for Scientific Research–Flanders (Belgium), EOS Project no 30468160 and project G0B0123N.

CONFLICT OF INTEREST

The authors declare no potential conflict of interest.

REFERENCES

1. N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, *Manopt, a Matlab toolbox for optimization on manifolds*, Journal of Machine Learning Research **15** (2014), no. 42, 1455–1459. URL https://www.manopt.org.

2. P. Breiding, F. Gesmundo, M. Michałek, and N. Vannieuwenhoven, *Algebraic compressed sensing*, Applied and Computational Harmonic Analysis **65** (2023), 374–406. URL https://www.sciencedirect.com/science/article/pii/S1063520323000271.
3. S. Budzinskiy and N. Zamarashkin, *Tensor train completion: local recovery guarantees via Riemannian optimization*, Numerical Linear Algebra with Applications **30** (2023), no. 6, e2520. URL https://doi.org/10.1002/nla.2520.
4. J.-F. Cai, W. Huang, H. Wang, and K. Wei, *Tensor completion via tensor train based low-rank quotient geometry under a preconditioned metric*, arXiv preprint (2022). URL https://doi.org/10.48550/arXiv.2209.04786.
5. J. D. Carroll and J. J. Chang, *Analysis of individual differences in multidimensional scaling via an n-way generalization of Eckart-Young decomposition*, Psychometrika **35** (1970), 283–319. URL https://api.semanticscholar.org/CorpusID:50364581.
6. L. De Lathauwer, B. De Moor, and J. Vandewalle, *A multilinear singular value decomposition*, SIAM Journal on Matrix Analysis and Applications **21** (2000), no. 4, 1253–1278.
7. B. Gao and P.-A. Absil, *A Riemannian rank-adaptive method for low-rank matrix completion*, Computational Optimization and Applications **81** (2022), 67–90. URL https://doi.org/10.1007/s10589-021-00328-w.
8. L. Grasedyck, D. Kressner, and C. Tobler, *A literature survey of low-rank tensor approximation techniques*, GAMM-Mitteilungen **36** (2013), no. 1, 53–78. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/gamm.201310004.
9. W. Hackbusch, *Tensor Spaces and Numerical Tensor Calculus*, *Springer Series in Computational Mathematics*, vol. 56, 2nd edn., Springer Cham, 2019, . URL https://doi.org/10.1007/978-3-642-28027-6.
10. R. A. Harshman, *Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-model factor analysis*, 1970, 1–84. URL https://api.semanticscholar.org/CorpusID:6816804.
11. C. J. Hillar and L.-H. Lim, *Most tensor problems are NP-hard*, Journal of the ACM **60** (2013), no. 6. URL https://doi.org/10.1145/2512329.
12. S. Holtz, T. Rohwedder, and R. Schneider, *The alternating linear scheme for tensor train optimization in the tensor train format*, SIAM Journal on Scientific Computing **34** (2012), no. 2, A683–A713. URL https://doi.org/10.1137/100818893.
13. S. Holtz, T. Rohwedder, and R. Schneider, *On manifolds of tensors of fixed TT-rank*, Numerische Mathematik **120** (2012), no. 4, 701–731. URL https://doi.org/10.1007/s00211-011-0419-7.
14. H. Kasai and B. Mishra, *Low-rank tensor completion: a Riemannian manifold preconditioning approach*, M. F. Balcan and K. Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, *Proceedings of Machine Learning Research*, vol. 48, PMLR, New York, New York, USA, 2016, 1012–1021. URL https://proceedings.mlr.press/v48/kasai16.html.
15. C.-Y. Ko, K. Batselier, L. Daniel, W. Yu, and N. Wong, *Fast and accurate tensor completion with total variation regularized tensor trains*, IEEE Transactions on Image Processing **29** (2020), 6918–6931. URL https://doi.org/10.1109/TIP.2020.2995061.
16. D. Kressner, M. Steinlechner, and A. Uschmajew, *Low-rank tensor methods with subspace correction for symmetric eigenvalue problems*, SIAM J. Sci. Comput. **36** (2014), no. 5, A2346–A2368. URL https://doi.org/10.1137/130949919.
17. D. Kressner, M. Steinlechner, and B. Vandereycken, *Low-rank tensor completion by Riemannian optimization*, BIT Numerical Mathematics **54** (2014), 447–468. URL https://doi.org/10.1007/s10543-013-0455-z.
18. B. Kutschan, *Tangent cones to tensor train varieties*, Linear Algebra and its Applications **544** (2018), 370–390. URL https://doi.org/10.1016/j.laa.2018.01.012.
19. B. Kutschan, *Convergence of gradient methods on hierarchical tensor varieties*, Ph.D. thesis, TU Berlin, 2019.
20. E. Levin, J. Kileel, and N. Boumal, *Finding stationary points on bounded-rank matrices: A geometric hurdle and a smooth remedy*, Mathematical Programming (2022). URL https://doi.org/10.1007/s10107-022-01851-2.
21. C. Lubich, I. V. Oseledets, and B. Vandereycken, *Time integration of tensor trains*, SIAM Journal on Numerical Analysis **53** (2015), no. 2, 917–941. URL https://doi.org/10.1137/140976546.
22. I. V. Oseledets, *Tensor-train decomposition*, SIAM Journal on Scientific Computing **33** (2011), no. 5, 2295–2317. URL https://doi.org/10.1137/090752286.
23. M. Psenka and N. Boumal, *Second-order optimization for tensors with fixed tensor-train rank*, arXiv preprint (2020). URL https://doi.org/10.48550/arXiv.2011.13395.
24. R. Schneider and A. Uschmajew, *Convergence results for projected line-search methods on varieties of low-rank matrices via Łojasiewicz inequality*, SIAM Journal on Optimization **25** (2015), no. 1, 622–646. URL https://doi.org/10.1137/140957822.
25. C. Schwab and C. J. Gittelson, *Sparse tensor discretizations of high-dimensional parametric and stochastic pdes*, Acta Numerica **20** (2011), 291–467.
26. M. Steinlechner, *Riemannian optimization for high-dimensional tensor completion*, SIAM Journal on Scientific Computing **38** (2016), no. 5, S461–S484. URL https://doi.org/10.1137/15M1010506.
27. M. M. Steinlechner, *Riemannian Optimization for Solving High-Dimensional Problems with Low-Rank Tensor Structure*, PhD Thesis **6958** (2016).
28. M. M. Steinlechner, *Riemannian optimization for solving high-dimensional problems with low-rank tensor structure*, Ph.D. thesis, MATHICSE, Lausanne, 2016, . URL https://doi.org/10.5075/epfl-thesis-6958.
29. L. R. Tucker, *The extension of factor analysis to three-dimensional matrices*, H. Gulliksen and N. Frederiksen (eds.), *Contributions to mathematical psychology.*, Holt, Rinehart and Winston, New York, 1964. 110–127.
30. L. R. Tucker, *Some mathematical notes on three-mode factor analysis*, Psychometrika **31** (1966), no. 3, 279–311.
31. A. Uschmajew and B. Vandereycken, *Greedy rank updates combined with riemannian descent methods for low-rank optimization*, *2015 International Conference on Sampling Theory and Applications (SampTA)*, 2015, 420–424, .
32. B. Vandereycken, *Low-rank matrix completion by riemannian optimization*, SIAM Journal on Optimization **23** (2013), 1214–1236.
33. C. Vermeylen, G. Olikier, and M. Van Barel, *An approximate projection onto the tangent cone to the variety of third-order tensors of bounded tensor-train rank*, F. Nielsen and F. Barbaresco (eds.), *Geometric Science of Information.*, Springer Nature Switzerland, Cham, 2023, 484–493. URL https://doi.org/10.1007/978-3-031-38271-0_48.
34. C. Vermeylen, G. Olikier, P.-A. Absil, and M. Van Barel, *Rank estimation for third-order tensor completion in the tensor-train format*, *31st European Signal Processing Conference (EUSIPCO)*, 2023, 965–969, . URL https://doi.org/10.23919/EUSIPCO58844.2023.10289827.
35. N. Vervliet, O. Debals, and L. De Lathauwer, *Exploiting efficient representations in large-scale tensor decompositions*, SIAM Journal on Scientific Computing **41** (2019), no. 2, A789–A815. URL https://doi.org/10.1137/17M1152371.

36. G. Zhou, W. Huang, K. A. Gallivan, P. Van Dooren, and P.-A. Absil, *A Riemannian rank-adaptive method for low-rank optimization*, Neurocomputing **192** (2016), 72–80. URL https://doi.org/10.1016/j.neucom.2016.02.030.

DATA AVAILABILITY STATEMENT

Our implementation of the RRAM – Algorithm 4 – and the scripts to regenerate the experiments are publicly available [‡].

How to cite this article: Vermeylen C., and Van Barel M.. A Riemannian rank-adaptive method for higher-order tensor completion in the tensor-train format. *J Numer Linear Algebra Appl.* 2024;00(00):1–18.

APPENDIX

A ORTHOGONAL PROJECTIONS

Projections onto vector spaces are frequently used in this work. More specifically, the proof of the angle condition in Theorem 2 relies on the following basic result.

Lemma 2. *Let $A \in \mathbb{R}^{n\times m}$ have rank r. If $\hat{A} = \hat{U}\hat{S}\hat{V}^\top$ is a truncated SVD of rank s of A, with $s < r$, then, for all $U \in \mathrm{St}(s,n)$ and all $V \in \mathrm{St}(s,m)$,*

$$\left\|P_{\hat{U}}A\right\| \geq \|P_U A\|, \qquad \left\|P_{\hat{U}}A\right\|^2 \geq \frac{s}{r}\|A\|^2, \tag{A1}$$

$$\left\|AP_{\hat{V}}\right\| \geq \|AP_V\|, \qquad \left\|AP_{\hat{V}}\right\|^2 \geq \frac{s}{r}\|A\|^2. \tag{A2}$$

Proof. By the Eckart–Young theorem, $\hat{A}$ is a projection of A onto

$$\mathbb{R}^{n\times m}_{\leq s} := \left\{X \in \mathbb{R}^{n\times m} \;\middle|\; \mathrm{rank}(X) \leq s\right\}.$$

Thus, since $\mathbb{R}^{n\times m}_{\leq s}$ is a closed cone, (28) holds. Moreover, since $\hat{S}\hat{V}^\top = \hat{U}^\top A$ and thus $\hat{A} = \hat{U}\hat{U}^\top A = P_{\hat{U}}A$, it holds that

$$\|P_{\hat{U}}A\|^2 = \max\left\{\|A_1\|^2 \;\middle|\; A_1 \in \mathbb{R}^{n\times m}_{\leq s},\ \langle A_1, A\rangle = \|A_1\|^2\right\}.$$

Furthermore, for all $U \in \mathrm{St}(s,n)$, $\langle P_U A, A\rangle = \left\langle P_U A, P_U A + P_U^\perp A\right\rangle = \|P_U A\|^2$. Hence,

$$\left\{P_U A \mid U \in \mathrm{St}(s,n)\right\} \subseteq \left\{\|A_1\|^2 \;\middle|\; A_1 \in \mathbb{R}^{n\times m}_{\leq s},\ \langle A_1, A\rangle = \|A_1\|^2\right\}.$$

Thus, $\|P_{\hat{U}}A\|^2 = \max\limits_{U\in\mathrm{St}(s,n)} \|P_U A\|^2$. The left inequality in (A1) follows, and the one in (A2) can be obtained similarly.

By orthogonal invariance of the Frobenius norm and by definition of $\hat{A}$,

$$\|A\|^2 = \sum_{i=1}^{r}\sigma_i^2, \qquad \|\hat{A}\|^2 = \sum_{i=1}^{s}\sigma_i^2,$$

where $\sigma_1, \ldots, \sigma_r$ are the singular values of A in decreasing order. Moreover, either $\sigma_s^2 \geq \frac{1}{r}\sum_{i=1}^{r}\sigma_i^2$ or $\sigma_s^2 < \frac{1}{r}\sum_{i=1}^{r}\sigma_i^2$. In the first case, we have

$$\|\hat{A}\|^2 = \sum_{i=1}^{s}\sigma_i^2 \geq s\sigma_s^2 \geq s\frac{\sum_{i=1}^{r}\sigma_i^2}{r} = \frac{s}{r}\|A\|^2.$$

In the second case, we have

$$\|\hat{A}\|^2 = \sum_{i=1}^{r}\sigma_i^2 - \sum_{i=s+1}^{r}\sigma_i^2 \geq \|A\|^2 - (r-s)\sigma_s^2 > \|A\|^2 - (r-s)\frac{\sum_{i=1}^{r}\sigma_i^2}{r} = \frac{s}{r}\|A\|^2.$$

Thus, in both cases, the second inequality in (A1) holds. The second inequality in (A2) can be obtained in a similar way. □

[‡] https://github.com/CharlotteVermeylen/RRAM_TT_completion

ORCID

Charlotte Vermeylen https://orcid.org/0000-0002-3615-6571

Marc Van Barel https://orcid.org/0000-0002-7372-382X