

## Second-order optimization for tensors with fixed tensor-train rank

**Michael Psenka**

*Princeton University, Princeton, NJ, USA*

MPSENKA@PRINCETON.EDU

**Nicolas Boumal**

*EPFL, Switzerland*

NICOLAS.BOUMAL@EPFL.CH

### Abstract

There are several different notions of “low rank” for tensors, associated to different formats. Among them, the Tensor Train (TT) format is particularly well suited for tensors of high order, as it circumvents the curse of dimensionality: an appreciable property for certain high-dimensional applications. It is often convenient to model such applications as optimization over the set of tensors with fixed (and low) TT rank. That set is a smooth manifold. Exploiting this fact, others have shown that Riemannian optimization techniques can perform particularly well on tasks such as tensor completion and special large-scale linear systems from PDEs. So far, however, these optimization techniques have been limited to first-order methods, likely because of the technical hurdles in deriving exact expressions for the Riemannian Hessian. In this paper, we derive a formula and efficient algorithm to compute the Riemannian Hessian on this manifold. This allows us to implement second-order optimization algorithms (namely, the Riemannian trust-region method) and to analyze the conditioning of optimization problems over the fixed TT rank manifold. In settings of interest, we show improved optimization performance on tensor completion compared to first-order methods and alternating least squares (ALS). Our work could have applications in training of neural networks with tensor layers. Our code is freely available.

### 1. Introduction

Tensors of order  $d$  are multi-dimensional arrays with some size  $n_1 \times \dots \times n_d$ . They occur in numerous applications, sometimes as the unknown in an optimization problem. Aside from training neural networks, examples include predicting gene expression (e.g. [8]) and solving differential equations (e.g. [6]).

In the same way that optimizing over large matrices (tensors of order two) may be challenging, so optimizing over large tensors requires care. When optimizing over matrices, it is often the case that one can meaningfully restrict attention to matrices of a given low rank  $r$ . This may be either because the solution of the problem genuinely is a matrix of rank  $r$ , or because it can be well approximated by one. When it comes to tensors, there exist several notions of low rank, with their pros and cons. We focus on the notion of rank associated with the *tensor train* (TT) format (also known as *matrix product state* (MPS) in the physics community).

The *TT-rank* of a tensor  $X$  of order  $d$  is a tuple of integers:  $\text{rank}_{\text{TT}}(X) = \mathbf{r} = (r_1, \dots, r_{d-1})$ . For  $d = 2$ , it reduces to the usual notion of matrix rank. We consider optimization problems over the set

$$\mathcal{M}_{\mathbf{r}} = \{X \in \mathbb{R}^{n_1 \times \dots \times n_d} : \text{rank}_{\text{TT}}(X) = \mathbf{r}\}.$$

As reviewed below, this set can be endowed with the structure of a Riemannian submanifold of the Euclidean space  $\mathbb{R}^{n_1 \times \dots \times n_d}$ . This makes it possible to use general techniques from Riemannian optimization [2] to minimize functions on  $\mathcal{M}_r$ .

Other authors have exploited the Riemannian structure of  $\mathcal{M}_r$  to design first-order optimization algorithms such as gradient descent and certain quasi-Newton schemes [14, 15]. However, no second-order optimization algorithms on  $\mathcal{M}_r$  have been implemented yet. Having access to the Riemannian Hessian, we can expect to see superlinear local convergence of algorithms such as the Riemannian trust-region method [1]. This is manifest in our numerical experiments.

Motivated by the role of tensors in modern machine learning applications (including as a means to encode weights in layers of neural networks) and by the recently revived interest in second-order methods for machine learning tasks,<sup>1</sup> in this paper we derive the geometric and numerical tools necessary to implement second-order optimization algorithms on  $\mathcal{M}_r$ . In particular, we implement the tools necessary to use Riemannian trust-region methods [1] in the Manopt toolbox [5]. The main ingredient is an efficient procedure to evaluate Riemannian Hessians.

To illustrate the benefits of second-order optimization algorithms on  $\mathcal{M}_r$ , we experiment with low-rank tensor completion, analogous to the well-known low-rank matrix completion problem.

An alternative encoding for tensors is the *Tucker format*, with its associated notion of Tucker or multilinear rank. The set of tensors with fixed Tucker rank is also a manifold. Second-order methods for optimization over that manifold are developed in [7]. For high-order tensors, the strength of the TT format is that the dimension of  $\mathcal{M}_r$  grows linearly in  $d$ , whereas the dimension of the manifold of fixed Tucker rank grows exponentially in  $d$  (the base of this exponential growth is the rank, not the size of the tensors). The canonical polyadic (CP) format also escapes the curse of dimensionality, but the set of tensors with fixed CP rank is difficult to handle for optimization [15, §9.3].

## 2. Tensor train format

Suppose we can factor a tensor  $X$  of order  $d$  and size  $n_1 \times \dots \times n_d$  into the following form:

$$X(i_1, \dots, i_d) = U_1(i_1)U_2(i_2) \cdots U_d(i_d) \tag{1}$$

where each  $U_\mu(i_\mu) \in \mathbb{R}^{r_{\mu-1} \times r_\mu}$  is a small matrix for some integers  $r_0, \dots, r_d$ . Since  $X(i_1, \dots, i_d)$  is a scalar, we necessarily have  $r_0 = r_d = 1$ : we collect the remaining sizes in the tuple  $\mathbf{r} = (r_1, \dots, r_{d-1})$ . For each  $\mu \in [d] := \{1, \dots, d\}$ , we stack all of the matrices  $U_\mu(1), \dots, U_\mu(n_\mu)$  together to form a third-order tensor  $U_\mu \in \mathbb{R}^{r_{\mu-1} \times n_\mu \times r_\mu}$ . These third-order tensors are called the *cores*, and the set of cores  $\{U_1, \dots, U_d\}$  form a *tensor train decomposition* of  $X$ . The size  $\mathbf{r}$  of the decomposition is  $(r_1, \dots, r_{d-1})$ , a vector closely related to the *TT-rank* defined below. A tensor decomposed into this format is a *TT-tensor* [11]. For example, a TT-tensor of order  $d$  and size  $n \times \dots \times n$  with “constant”  $\mathbf{r} = (r, \dots, r)$  is fully specified by  $O(r^2nd)$  real numbers (as opposed to  $n^d$  in full generality): the linear scaling in  $d$  (as opposed to exponential) is how this format escapes the curse of dimensionality [12]. Of course, the decomposition is not unique.

We define the TT-rank of a tensor  $Z \in \mathbb{R}^{n_1 \times \dots \times n_d}$  by the following:

$$\text{rank}_{\text{TT}}(Z) := \left( \text{rank}(Z^{<1>}), \dots, \text{rank}(Z^{<d-1>}) \right),$$

---

1. See for example the program of the NeurIPS 2019 workshop “Beyond first order methods in machine learning systems,” <https://sites.google.com/site/optneurips19/>.

where each  $Z^{<k>}$  is a so-called *flattening* of the tensor into a matrix (see Def. 4 in Appendix B.1).

The following theorem provides support for the latter definition. In essence, it states that  $Z$  admits a TT-decomposition with size  $\mathbf{r} = \text{rank}_{\text{TT}}(Z)$ , but not less. See [15, Thm. 9.2] for a proof.

**Theorem 1** *Let  $Z \in \mathbb{R}^{n_1 \times \dots \times n_d}$ ,  $Z \neq 0$ . Denote  $\tilde{r}_k := \text{rank}(Z^{<k>})$ . For any tensor train decomposition  $U_1, \dots, U_d$  of size  $\mathbf{r} = (r_1, \dots, r_{d-1})$ , it necessarily holds that  $r_k \geq \tilde{r}_k$  for all  $k \in [d-1]$ , and it is furthermore possible to obtain a decomposition such that equality holds.*

Based on the latter statement, we say a TT decomposition of a tensor is *minimal* if the sizes of the cores match the TT-rank of the tensor.

### 3. Smooth manifold structure

We give a concise overview of the geometry of  $\mathcal{M}_{\mathbf{r}}$  restricted to properties useful for optimization. For book-length introductions to the topic of Riemannian optimization, we direct the reader to [2, 4]. For a full treatment of the geometry of fixed TT-rank tensors specifically, we recommend [10, 14, 15].

The set of tensors of size  $n_1 \times \dots \times n_d$  and fixed TT-rank  $\mathbf{r} = (r_1, \dots, r_{d-1})$ ,

$$\mathcal{M}_{\mathbf{r}} = \{X \in \mathbb{R}^{n_1 \times \dots \times n_d} : \text{rank}_{\text{TT}}(X) = \mathbf{r}\},$$

is a subset of  $\mathbb{R}^{n_1 \times \dots \times n_d}$ . In the same way that the set of matrices of size  $m \times n$  and rank  $r$  is smoothly embedded in  $\mathbb{R}^{m \times n}$ ,  $\mathcal{M}_{\mathbf{r}}$  is a smooth embedded submanifold of  $\mathbb{R}^{n_1 \times \dots \times n_d}$  of dimension

$$\dim \mathcal{M}_{\mathbf{r}} = \sum_{i=1}^d r_{i-1} n_i r_i - \sum_{i=1}^{d-1} r_i^2. \quad (2)$$

This means that around each point  $X \in \mathcal{M}_{\mathbf{r}}$  we can define a linearization of  $\mathcal{M}_{\mathbf{r}}$  called the *tangent space* at  $X$ . This is a linear subspace  $T_X \mathcal{M}_{\mathbf{r}}$  of  $\mathbb{R}^{n_1 \times \dots \times n_d}$  which consists in all the “vectors” (in fact, tensors) of the form  $c'(0)$  where  $c(t)$  is a smooth curve in  $\mathbb{R}^{n_1 \times \dots \times n_d}$  which lies entirely on  $\mathcal{M}_{\mathbf{r}}$  and passes through  $X$  so that  $c(0) = X$ . Explicitly, given a minimal left-orthogonal decomposition  $U_1, \dots, U_d$  of  $X$  (left-orthogonal is defined in Appendix B.2), we can parametrize a tangent vector  $V$  at  $X$  by tensors  $\delta V_1, \dots, \delta V_d$  of the same shape as  $U_1, \dots, U_d$  such that

$$V(i_1, \dots, i_d) = \sum_{k=1}^d U_1(i_1) \cdots U_{k-1}(i_{k-1}) \delta V_k(i_k) U_{k+1}(i_{k+1}) \cdots U_d(i_d) \quad (3)$$

and  $(\delta V_i^L)^\top U_i^L = 0$  for  $i \in [d-1]$ , where  $U_i^L := (U_i)^{<2>}$  and  $\delta V_i^L := (\delta V_i)^{<2>}$  (see Def. 5 of Appendix B.1). The space  $\mathbb{R}^{n_1 \times \dots \times n_d}$  has the usual Euclidean inner product

$$\langle V, W \rangle = \sum_{i_1, \dots, i_d} V(i_1, \dots, i_d) W(i_1, \dots, i_d).$$

We equip each tangent space with the same inner product simply by restricting the domain. This turns  $\mathcal{M}_{\mathbf{r}}$  into a Riemannian manifold; specifically: a Riemannian submanifold of  $\mathbb{R}^{n_1 \times \dots \times n_d}$ .

A function  $f: \mathcal{M}_{\mathbf{r}} \rightarrow \mathbb{R}$  is smooth if and only if it is the restriction of a smooth function  $\bar{f}$  defined on a neighborhood of  $\mathcal{M}_{\mathbf{r}}$  in  $\mathbb{R}^{n_1 \times \dots \times n_d}$ . The Riemannian structure affords us a notion of

gradient and Hessian for  $f$ , of central importance for optimization. Specifically, the *Riemannian gradient* of  $f$  at  $X \in \mathcal{M}_r$  is the (unique) tangent vector  $\text{grad}f(X) \in \mathbb{T}_X \mathcal{M}_r$  such that

$$\forall V \in \mathbb{T}_X \mathcal{M}_r, \quad \langle \text{grad}f(X), V \rangle = \lim_{t \rightarrow 0} \frac{\bar{f}(X + tV) - \bar{f}(X)}{t}.$$

It can be shown that this does not depend on the choice of smooth extension  $\bar{f}$ . If  $\mathcal{P}_X$  denotes the orthogonal projector from  $\mathbb{R}^{n_1 \times \dots \times n_d}$  to  $\mathbb{T}_X \mathcal{M}_r$ , it is easy to verify that

$$\text{grad}f(X) = \mathcal{P}_X(\partial \bar{f}(X)),$$

where  $\partial \bar{f}(X)$  is the (classical) gradient of  $\bar{f}$  at  $X$ . Provided  $\partial \bar{f}(X)$  is sufficiently structured, this can be computed efficiently.

A *retraction* is a smooth map on the tangent bundle which provides maps  $R_X: \mathbb{T}_X \mathcal{M}_r \rightarrow \mathcal{M}_r$  such that  $c(t) = R_X(tV)$  is a smooth curve on  $\mathcal{M}_r$  satisfying  $c(0) = X$  and  $c'(0) = V$ . For example, a computationally favorable choice for  $\mathcal{M}_r$  is the *TT-SVD* [11],[15, §9.3.4].

Combined, the tools described here are sufficient to develop first-order optimization methods on  $\mathcal{M}_r$ , including Riemannian gradient descent and even some quasi-Newton methods [14]. However, to implement true-to-form second-order optimization methods, we also need access to the Riemannian Hessian: this is our main object of study. But first, we need a second look at tangent vectors.

**An alternative parametrization of  $\mathbb{T}_X \mathcal{M}_r$ .** Orthogonal projections and inner products of tangent vectors are computed frequently in optimization algorithms, so it is key to have a parametrization of  $\mathbb{T}_X \mathcal{M}_r$  that yields efficient computation of both. The following parametrization of  $\mathbb{T}_X \mathcal{M}_r$  was first proposed in [9], and further elaborated on in [14]. Given a tangent vector represented by  $\delta V_1, \dots, \delta V_d$  (as in eq. (3)), we can generate another representation  $\delta \tilde{V}_1, \dots, \delta \tilde{V}_d$  such that  $(\delta \tilde{V}_i^L)^\top U_i^L = 0$  for  $i \in [d-1]$  and such that the tangent vector  $V$  is given by:

$$V(i_1, \dots, i_d) = \sum_{k=1}^d U_1(i_1) \cdots U_{k-1}(i_{k-1}) \delta \tilde{V}_k(i_k) \tilde{U}_{k+1}(i_{k+1}) \cdots \tilde{U}_d(i_d) \quad (4)$$

where  $\{\tilde{U}_k\}$  are the right-orthogonalized cores from  $\{U_k\}$  (see Appendix B.2). Conversely, we can also recover  $\delta V_1, \dots, \delta V_d$  from  $\delta \tilde{V}_1, \dots, \delta \tilde{V}_d$ . The inner product of two tangent vectors  $V, W$  with parametrizations  $\delta \tilde{V}_1, \dots, \delta \tilde{V}_d$ , and  $\delta \tilde{W}_1, \dots, \delta \tilde{W}_d$  admits a convenient expression:

$$\langle V, W \rangle = \sum_{i=1}^d \langle \delta \tilde{V}_i, \delta \tilde{W}_i \rangle.$$

This is computable in  $O(dnr^2)$  flops, where  $r := \max_\mu r_\mu$  and  $n := \max_\mu n_\mu$ . Importantly for our purpose, the tangent space can be decomposed into  $d$  orthogonal subspaces [14], so we can decompose the orthogonal projector  $\mathcal{P}_X$  into  $d$  orthogonal components:

$$\mathcal{P}_X = \mathcal{P}_X^1 + \cdots + \mathcal{P}_X^d \quad (5)$$

where  $\mathcal{P}_X^1, \dots, \mathcal{P}_X^d$  are the projectors to the orthogonal subspaces and are given by the following:

$$(\mathcal{P}_X^k(Z))^{\langle k \rangle} = (I_{n_k} \otimes X_{\leq k-1}) \left( (I_{n_k r_{k-1}} - U_k^L (U_k^L)^\top) (I_{n_k} \otimes X_{\leq k-1}^\top) Z^{\langle k \rangle} \tilde{X}_{\geq k+1} \right) \tilde{X}_{\geq k+1}^\top \quad (6)$$

for  $k \in [d - 1]$ , and

$$(\mathcal{P}_X^d(Z))^{\langle d \rangle} = (I_{n_d} \otimes X_{\leq d-1})(I_{n_d} \otimes X_{\leq d-1}^\top)Z^{\langle d \rangle}, \quad (7)$$

where  $X_{\leq k-1}$ ,  $\tilde{X}_{\geq k+1}$  are so-called *interface matrices* from decompositions  $\{U_k\}$  and  $\{\tilde{U}_k\}$  respectively (see Def. 4 in Appendix B.1). We use parametrization  $\{\delta V_k\}$  to represent tangent vectors in differentials, and  $\{\delta \tilde{V}_k\}$  for the resulting tangent vector after orthogonal projection to the tangent space. We prove interchangeability between these parametrizations in Lemma 12 of Appendix C.

#### 4. Riemannian Hessian

The Riemannian Hessian of  $f$  at  $X$ —a symmetric operator to and from  $T_X \mathcal{M}_r$ —admits an explicit expression in terms of the Euclidean derivatives of  $\bar{f}$  at  $X$ . It is shown in [3] for general Riemannian submanifolds that, for all  $V \in T_X \mathcal{M}_r$ ,

$$\text{Hess}f(X)[V] = \mathcal{P}_X \partial^2 \bar{f}(X)[V] + \mathcal{P}_X (D_V \mathcal{P}_X) \partial \bar{f}(X). \quad (8)$$

A few comments are in order. For the first term,  $\partial^2 \bar{f}(X)[V]$  is the Euclidean Hessian of  $\bar{f}$  at  $X$  along  $V$ , the result of which is then projected to  $T_X \mathcal{M}_r$  through  $\mathcal{P}_X$ . The second term is a “correction term” in the sense that it modifies the (projected) Euclidean Hessian to capture the Riemannian geometry of  $\mathcal{M}_r$ . The notation  $D_V \mathcal{P}_X$  denotes the differential of the map  $X \mapsto \mathcal{P}_X$  at  $X$  along the direction  $V$ , so that

$$(D_V \mathcal{P}_X)Z = D_V(\mathcal{P}_X Z) = \lim_{t \rightarrow 0} \frac{\mathcal{P}_{c(t)}(Z) - \mathcal{P}_{c(0)}(Z)}{t},$$

where  $c(t)$  is any smooth curve on  $\mathcal{M}_r$  such that  $c(0) = X$  and  $c'(0) = V$ . In words: it is the derivative of the orthogonal projector to  $T_X \mathcal{M}_r$  as we perturb  $X$  along the tangent direction  $V$ . As such,  $D_V \mathcal{P}_X$  is itself a linear operator from  $\mathbb{R}^{n_1 \times \dots \times n_d}$  to  $\mathbb{R}^{n_1 \times \dots \times n_d}$ . As shown in [3], the correction term depends only on the normal component of  $\partial \bar{f}(X)$ . Moreover, the operation which maps a tangent vector  $V$  and a normal vector  $N$  to the tangent vector  $\mathcal{P}_X (D_V \mathcal{P}_X)N$  is the *Weingarten map*: a standard object in geometry.

Note that splitting  $\mathcal{P}_X = \sum_{i=1}^d \mathcal{P}_X^i$  as in eq. (5) allows us to rewrite the correction term as:

$$\mathcal{P}_X (D_V \mathcal{P}_X)(Z) = \sum_{k=1}^d \mathcal{P}_X^k (D_V \mathcal{P}_X^k)(Z) + \sum_{i=1}^d \sum_{j=1, j \neq i}^d \mathcal{P}_X^i (D_V \mathcal{P}_X^j)(Z). \quad (9)$$

The double sum would seem to take too many flops to compute. However, we show in this paper an expression for these “cross-terms”  $\mathcal{P}_X^i (D_V \mathcal{P}_X^j)(Z)$ ,  $i \neq j$  in a way that yields a computation of the whole double sum in virtually no extra flops after computing  $\sum_{k=1}^d \mathcal{P}_X^k (D_V \mathcal{P}_X^k)(Z)$ , which we do efficiently.

We now present the main contribution of this paper: simplified formulas for the correction term that yield computationally efficient algorithms for the Hessian. Proofs for these formulas can be found in Appendix D, and time complexity analyses can be found in Appendix E. See Appendix B.2 for definitions of the small invertible matrices  $R_k \in \mathbb{R}^{r_k \times r_k}$  and matrices  $\tilde{X}_{\geq k+1} \in \mathbb{R}^{n_{k+1} \dots n_d \times r_k}$ . We introduce the matrices  $V_{<k}$  and  $V_{>k}$  as the *variational interface matrices* of tangent vector  $V$ , defined in Def. 7 of Appendix B.1.

**Theorem 2** *The terms in the first sum of eq. (9) can be computed as follows. For  $k < d$ , we have:*

$$\begin{aligned} (\mathcal{P}_X^k \mathbf{D}_V \mathcal{P}_X^k Z)^{\langle k \rangle} &= (I \otimes X_{\langle k \rangle}) \left( ((I - U_k^L (U_k^L)^\top) (I \otimes V_{\langle k \rangle}^\top)) Z^{\langle k \rangle} \tilde{X}_{\geq k+1}^\top \right) \tilde{X}_{\geq k+1}^\top \\ &\quad - (I \otimes X_{\langle k \rangle}) \left( \delta V_k^L (U_k^L)^\top (I \otimes X_{\langle k \rangle}) Z^{\langle k \rangle} \tilde{X}_{\geq k+1}^\top \right) \tilde{X}_{\geq k+1}^\top \\ &\quad + (I \otimes X_{\langle k \rangle}) ((I - U_k^L (U_k^L)^\top) (I_{n_k} \otimes X_{\leq k-1}^\top) Z^{\langle k \rangle} (I - \tilde{X}_{\geq k+1} \tilde{X}_{\geq k+1}^\top) V_{\geq k+1} R_k^{-1}) \tilde{X}_{\geq k+1}^\top \end{aligned}$$

while for  $k = d$  we have

$$(\mathcal{P}_X^d \mathbf{D}_V \mathcal{P}_X^d Z)^{\langle d \rangle} = (I \otimes X_{\leq d-1}) (I \otimes V_{\langle d \rangle}^\top) Z^{\langle d \rangle}.$$

For typically structured  $Z$  (e.g., full, sparse, low TT-rank), these formulas yield algorithms for computing the first sum in a number of arithmetic operations similar to that required for the computation of  $\mathcal{P}_X(Z)$ .

For the cross-terms (non-diagonal terms), we show the identity  $\mathcal{P}_X^i \mathbf{D}_V \mathcal{P}_X^j Z = -(\mathbf{D}_V \mathcal{P}_X^i) \mathcal{P}_X^j Z$ . This allows us to express the cross-terms in not only a simplified way, but into an expression that allows us to re-use computations already made for the diagonal terms:

**Theorem 3** *The terms in the double sum of eq. (9) can be computed as follows.*

$$\begin{aligned} j > i, i < d : (\mathcal{P}_X^i \mathbf{D}_V \mathcal{P}_X^j Z)^{\langle i \rangle} &= (I \otimes X_{\langle i \rangle}) \left( \delta V_i^L (Y_{\geq i+1}^j)^\top \tilde{X}_{> i}^\top \right) \tilde{X}_{> i}^\top, \\ j < i, i < d : (\mathcal{P}_X^i \mathbf{D}_V \mathcal{P}_X^j Z)^{\langle i \rangle} &= -(I \otimes X_{\langle i \rangle}) \left( (I - U_k^L (U_k^L)^\top) (I \otimes V_{\langle i \rangle}^\top) Y_{\leq i}^j \right) \tilde{X}_{\geq i+1}^\top, \\ j < i, i = d : (\mathcal{P}_X^d \mathbf{D}_V \mathcal{P}_X^j Z)^{\langle d \rangle} &= -(I \otimes X_{\langle d \rangle}) (I \otimes V_{\langle d \rangle}^\top) Y_{\leq d}^j, \end{aligned}$$

where  $Y^j$  is the TT-tensor given by  $\mathcal{P}_X^j Z$ . Given the set  $\{Y^1, \dots, Y^n\}$ , which are computed as a by-product from efficient algorithms for the first sum, the cross-terms  $\sum_{i=1}^d \sum_{j=1, j \neq i}^d \mathcal{P}_X^i (\mathbf{D}_V \mathcal{P}_X^j)(Z)$  are computable in  $O(d^2 n r^3)$  arithmetic operations.

## 5. Numerical analysis on tensor completion

Using our analytical expression of the Riemannian Hessian, we develop a Riemannian Trust Regions (RTR) method for solving optimization problems over  $\mathcal{M}_r$  [1, 5]. To assess the performance of this method, we compare RTR with Alternating Least Squares (ALS) and a conjugate gradient method on tensor completion (RTTC) [13], both of which were coded by Steinlechner et al. We also compare to RTR when we use a finite-difference approximation of the Riemannian Hessian: we denote the resulting algorithm FD-TR.

In our experiments, all tensors are of size  $(4, 4, \dots, 4)$  and some order  $d$ , specified at each experiment. For each experiment, we report the convergence of each algorithm in terms of cost (training cost), test cost from an independent set of samples, and gradient norm for algorithms where this applies (RTR, FD-TR, RTTC). A quantity of critical importance is the *oversampling ratio*:  $|\Omega|/\dim(\mathcal{M}_r)$ , where  $|\Omega|$  is the number of observed indices. We also report the *sampling ratio*,  $|\Omega|/4^d$ .

Graphs of the results and further details of the experiments can be found in Appendix A. In summary, we find that RTR is slower on versions of tensor completion with better conditioned Hessians, but outperforms other algorithms in more challenging instances of the problem where the target point Hessian has worse conditioning (something we can assess using our Hessian formulas).

## Funding

This work was supported by the National Science Foundation through award DMS-1719558.

## References

- [1] P.-A. Absil, C. G. Baker, and K. A. Gallivan. Trust-region methods on Riemannian manifolds. *Foundations of Computational Mathematics*, 7(3):303–330, 2007. doi: 10.1007/s10208-005-0179-9.
- [2] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, 2008. ISBN 978-0-691-13298-3.
- [3] P.-A. Absil, R. Mahony, and J. Trumpf. An extrinsic look at the Riemannian Hessian. In Frank Nielsen and Frédéric Barbaresco, editors, *Geometric Science of Information*, volume 8085 of *Lecture Notes in Computer Science*, pages 361–368. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-40019-3. doi: 10.1007/978-3-642-40020-9\_39.
- [4] N. Boumal. An introduction to optimization on smooth manifolds. Available online, May 2020. URL <http://www.nicolasboumal.net/book>.
- [5] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre. Manopt, a Matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research*, 15(42):1455–1459, 2014. URL <https://www.manopt.org>.
- [6] S.V. Dolgov. A tensor decomposition algorithm for large ODEs with conservation laws. *Computational Methods in Applied Mathematics*, 19(1):23 – 38, 2019. URL <https://www.degruyter.com/view/journals/cmam/19/1/article-p23.xml>.
- [7] G. Heide and V. Schulz. A Riemannian trust-region method for low-rank tensor completion. *Numerical Linear Algebra with Applications*, 25(6):e2175, 2018.
- [8] M. Iwata, L. Yuan, Q. Zhao, Y. Tabei, F. Berenger, R. Sawada, S. Akiyoshi, M. Hamano, and Y. Yamanishi. Predicting drug-induced transcriptome responses of a wide range of human cell lines by a novel tensor-train decomposition algorithm. *Bioinformatics*, 35(14):i191–i199, 07 2019. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz313. URL <https://doi.org/10.1093/bioinformatics/btz313>.
- [9] B. Khoromskij, I. Oseledets, and R. Schneider. Efficient time-stepping scheme for dynamics on tt-manifolds. 04 2012.
- [10] D. Kressner, M. Steinlechner, and B. Vandereycken. Low-rank tensor completion by Riemannian optimization. *BIT Numerical Mathematics*, 54(2):447–468, Jun 2014. doi: 10.1007/s10543-013-0455-z.
- [11] I. V. Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011. doi: 10.1137/090752286. URL <https://doi.org/10.1137/090752286>.

- [12] I.V. Oseledets and E.E. Tyrtshnikov. Breaking the curse of dimensionality, or how to use SVD in many dimensions. *SIAM Journal on Scientific Computing*, 31(5):3744–3759, January 2009. doi: 10.1137/090748330.
- [13] M. Steinlechner. Riemannian optimization for high-dimensional tensor completion. *SIAM J. Scientific Computing*, 38, 2016.
- [14] M. Steinlechner. *Riemannian optimization for solving high-dimensional problems with low-rank tensor structure*. phdthesis, EPFL, 2016.
- [15] A. Uschmajew and B. Vandereycken. Geometric methods on low-rank matrix and tensor manifolds. In *Handbook of Variational Methods for Nonlinear Geometric Data*, pages 261–313. Springer International Publishing, 2020. doi: 10.1007/978-3-030-31351-7\_9.



### Appendix A. Experiments on tensor completion

We consider target points randomly generated on the manifold by constructing normally distributed TT-cores. Observed entries are chosen according to some distribution  $\mathbf{p} = (p_1, p_2, p_3, p_4)$  such that, for each sample index  $(i_1, \dots, i_d)$ , each  $i_k$  is chosen at random from  $\{1, 2, 3, 4\}$  according to the distribution  $\mathbf{p}$  (not necessarily uniform). These experiments illustrate the observation that second-order methods perform better on “harder” versions of tensor completion.

For each figure, we plot the (training loss/test loss/gradient norm) over 10 trials, each trial with a different random initialization and target tensor. The three algorithms are compared on each trial, and the 10 trials are then plotted over each other in a single chart. The differences in the problem setting for each figure are described in the figure details.

For Trust Regions, we used a starting radius of 100 and a maximum radius of  $100 \cdot 2^{11}$ .

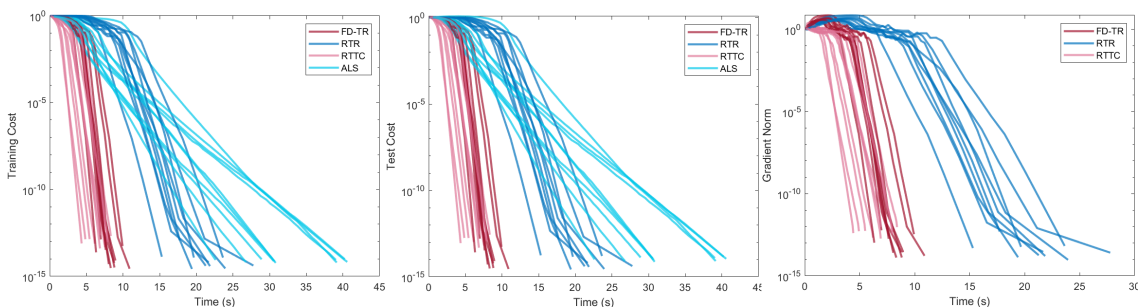


Figure 1:  $d = 9$ ,  $\mathbf{r} = (3, 5, 10, 10, 10, 10, 5, 3)$ , oversampling ratio: 20.5, sampling ratio: 0.1,  $\mathbf{p} = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$ , condition numbers of the Hessian at each target point: around 10. This could be considered as an “easier” tensor completion problem, as both the oversampling and sampling ratios are relatively high, and entries are sampled uniformly at random.

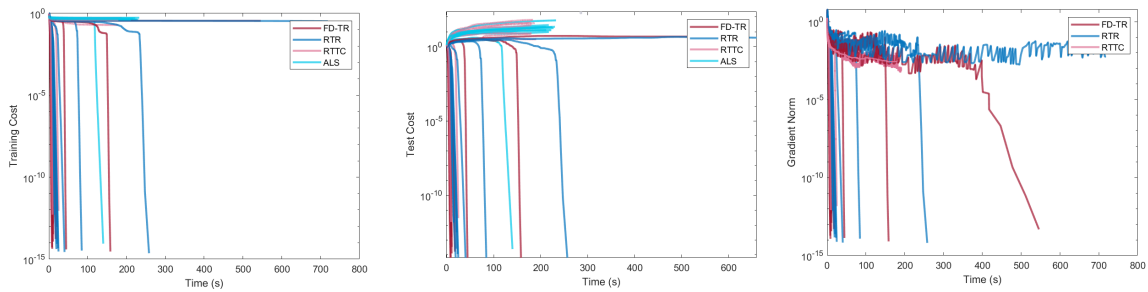


Figure 2:  $d = 9$ ,  $\mathbf{r} = (3, 4, 8, 12, 12, 8, 4, 3)$ , Oversampling Ratio: 5.2, Sampling ratio: 0.025,  $\mathbf{p} = (\frac{2}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})$ , Hessian condition numbers:  $10^2 - 10^3$ . As we decrease the oversampling ratio, the benefits of second-order methods start to show. In the allotted time, out of 10 trials, ALS converged once, RTTC converged twice, and both RTR and FD-TR converged 9 times. Note though that FD-TR converges faster than RTR; FD-TR performs similarly to RTR iteration-wise, but the analytic Hessian takes longer to compute than the finite difference approximation.

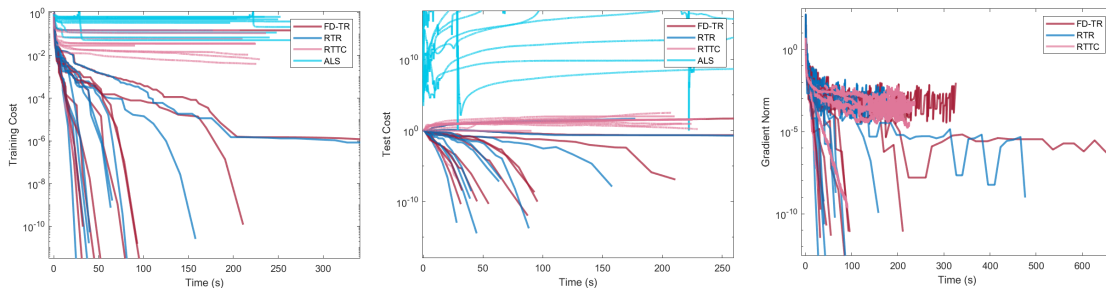


Figure 3:  $d = 9$ ,  $\mathbf{r} = (2, 2, 3, 3, 3, 3, 2, 2)$ , Oversampling Ratio: 5.1, Sampling ratio: 0.003,  $\mathbf{p} = (\frac{50}{65}, \frac{12}{65}, \frac{2}{65}, \frac{1}{65})$ , Hessian condition numbers:  $10^5 - 10^7$ . This example illustrates that having access to the true Hessian can yield an advantage over using a finite difference approximation. RTR outperformed all other algorithms in each instance separately.

## Appendix B. Proofs for the tensor train format

This section contains proofs for properties of tensors in the tensor train format. We use these initial results to work out the main result of the paper: the Riemannian Hessian for  $\mathcal{M}_{\mathbf{r}}$ .

### B.1. Preliminary definitions

In this subsection, we establish the various notation used in subsequent proofs. For any tensor  $X \in \mathbb{R}^{n_1 \times \dots \times n_d}$ , it holds that  $X \in \mathcal{M}_{\mathbf{r}}$  if and only if  $(\text{rank}(X^{<1>}), \dots, \text{rank}(X^{<d-1>})) = \mathbf{r}$ . For  $\mathcal{M}_{\mathbf{r}}$  to be non-empty, it is necessary and sufficient for  $r_{k-1} \leq n_k r_k$  and  $r_k \leq n_k r_{k-1}$  for all  $k \in [d]$ ,

so for the remainder of this section, we assume these conditions hold. This statement can be found in [15, eq. (9.32)].

For any  $X \in \mathcal{M}_r$ , unless otherwise stated, denote  $U_1, \dots, U_d$  to be a minimal, left-orthogonal TT-decomposition (note this decomposition is not unique), and let  $\tilde{U}_1, \dots, \tilde{U}_d$  be the resulting TT-decomposition from right-orthogonalization of  $U_1, \dots, U_d$  using [14, Alg. 4.1]. Let  $X_{\leq k}, X_{\geq k}$  denote the interface matrices for  $U_1, \dots, U_d$ , and let  $\tilde{X}_{\leq k}, \tilde{X}_{\geq k}$  denote the interface matrices for  $\tilde{U}_1, \dots, \tilde{U}_d$ .

**Definition 4** Let  $Z \in \mathbb{R}^{n_1 \times \dots \times n_d}$  be a tensor of order  $d$ . Then the “ $\mu$ th flattening”, written as  $Z^{<\mu>} \in \mathbb{R}^{n_1 \dots n_\mu \times n_{\mu+1} \dots n_d}$  flattens  $Z$  to a matrix of size  $(i_1 \dots i_\mu \times i_{\mu+1} \dots i_d)$ : within each dimension of the matrix, the indices are ordered colexicographically. (This is the same as calling Matlab’s “reshape” method on a multidimensional array with the specified target dimension.)

**Definition 5** Let  $U_\mu$  be a core of some TT-decomposition of  $X$ . Since cores are third-order tensors, there are only two non-vector  $\mu$ th flattenings:  $U_\mu^{<1>} \in \mathbb{R}^{r_{\mu-1} \times n_\mu \times r_\mu}$  and  $U_\mu^{<2>} \in \mathbb{R}^{r_{\mu-1} n_\mu \times r_\mu}$ , which we will denote  $U_\mu^R$  and  $U_\mu^L$  respectively. These are called the “right flattening” and “left flattening”. Without loss of generality, we assume that  $(U_\mu^L)^\top U_\mu^L = I_{r_\mu}$  for all  $\mu \in \{1, \dots, d-1\}$  (see [14, §4.2.1]).

**Definition 6** Let  $X$  be a TT-tensor with cores  $U_1, \dots, U_d$ . For each  $k \in \{0, \dots, d\}$ , define  $X_{\leq k} \in \mathbb{R}^{n_1 \dots n_k \times r_k}$  by the recursive formula  $X_{\leq k} = (I_{n_k} \otimes X_{\leq k-1})U_k^L$  and base case  $X_{\leq 0} = 1 \in \mathbb{R}$ . Similarly, for each  $k \in \{1, \dots, d+1\}$ , define  $X_{\geq k} \in \mathbb{R}^{r_{k-1} \times n_k \dots n_d}$  by the recursive formula  $X_{\geq k}^\top = U_k^R(X_{\geq k+1}^\top \otimes I_{n_k})$  and the base case  $X_{\geq d+1} = 1 \in \mathbb{R}$ . We call these matrices the interface matrices of  $X$ . While phrased differently, this definition is equivalent to the one given in [14, §4.1].

**Definition 7** Let  $V \in \mathbb{T}_X \mathcal{M}_r$  be represented as in eq. (3). For each  $k \in \{0, \dots, d\}$ , define  $V_{\leq k} \in \mathbb{R}^{n_1 \dots n_k \times r_k}$  by the recursive formula  $V_{\leq k} = (I_{n_k} \otimes V_{\leq k-1})U_k^L + (I_{n_k} \otimes X_{\leq k-1})\delta V_k^L$  and base case  $V_{\leq 0} = 0 \in \mathbb{R}$ . Similarly, for each  $k \in \{1, \dots, d+1\}$ , define  $V_{\geq k} \in \mathbb{R}^{r_{k-1} \times n_k \dots n_d}$  by the recursive formula  $V_{\geq k}^\top = U_k^R(V_{\geq k+1}^\top \otimes I_{n_k}) + \delta V_k^R(X_{\geq k+1}^\top \otimes I_{n_k})$  and the base case  $V_{\geq d+1} = 0 \in \mathbb{R}$ . We call these matrices the variational interface matrices of tangent vector  $V$ .

## B.2. $\mu$ -orthogonal decompositions

Let  $X \in \mathbb{R}^{n_1 \times \dots \times n_d}$  be a TT-tensor with a minimal decomposition  $U_\mu$ . We call a decomposition  $\mu$ -orthogonal if  $(U_k^L)^\top U_k^L = I_{r_k}$  for all  $k \in [\mu-1]$  and  $U_k^R(U_k^R)^\top = I_{r_{k-1}}$  for all  $k \in \{\mu+1, \dots, d\}$ . We also call  $d$ -orthogonal decompositions *left orthogonal* and 1-orthogonal decompositions *right orthogonal*. Transforming a decomposition to a  $\mu$ -orthogonal one without changing the underlying tensor is called  $\mu$ -orthogonalization, and any minimal TT-decomposition can be  $\mu$ -orthogonalized for any  $\mu \in [d]$  in  $O(dnr^3)$  flops using the algorithm presented in [14, Alg. 4.1].

We will frequently use the right-orthogonalization of a minimal left-orthogonal decomposition  $U_1, \dots, U_d$ ; denote these resulting right-orthogonal cores  $\tilde{U}_1, \dots, \tilde{U}_d$  and their interface matrices  $\tilde{X}_{\leq k}$  and  $\tilde{X}_{\geq k}$ . Note this is a different definition for  $\tilde{X}_{\geq k}$  from what is given for Theorem 2; these two definitions for  $\tilde{X}_{\geq k}$  indeed define the same matrix [14, §4.2.1]. We also define  $\{R_2, \dots, R_d\}$  by the relation  $X_{\geq k+1} = \tilde{X}_{\geq k+t}R_k$ , which are generated as a by-product of [14, Alg. 4.1].

### B.3. Proofs regarding the Tensor Train format

The following two lemmas can be found in [14, §4.1-2] and are fundamental identities for later proofs.

**Lemma 8** *Let  $X \in \mathcal{M}_r$ , and recall Definition 6. Let  $\mathbf{i} = (i_1, \dots, i_d)$  be a multi-index for  $X$ . The following two identities hold:*

1.  $\text{row}_{i_1 \dots i_k}(X_{\leq k}) = U_1(i_1) \cdots U_k(i_k)$
2.  $\text{row}_{i_1 \dots i_k}(X_{\geq k}) = (U_k(i_k) \cdots U_d(i_d))^\top$

where  $\text{row}_\mu(A) \in \mathbb{R}^{1 \times n}$  denotes the  $\mu$ th row of matrix  $A \in \mathbb{R}^{m \times n}$ . Lemma 8 is proven trivially from induction on the inductive definitions for the interface matrices. Using Lemma 8, it is straightforward to establish the following equation.

**Lemma 9** *For any TT-decomposition  $U_1, \dots, U_d$  of  $X$  (not necessarily left-orthogonal), the following identity holds:*

$$X^{<k>} = X_{\leq k} X_{\geq k+1}^\top$$

**Lemma 10** *If TT-cores  $U_1, \dots, U_d$  are left-orthogonal, the interface matrices  $X_{\leq 1}, \dots, X_{\leq d-1}$  have orthonormal columns:  $X_{\leq k}^\top X_{\leq k} = I_{r_k}$ . Similarly, the interface matrices  $\tilde{X}_{\geq 2}, \dots, \tilde{X}_{\geq d}$  have orthonormal columns.*

**Proof** We prove by induction, starting with the left-orthogonal case. For the base case, note that  $X_{\leq 1} = U_1^L$ , and we know that  $(U_1^L)^\top U_1^L = I_{r_1}$  by left-orthogonality, proving the base case. Now assuming that  $X_{\leq k-1}^\top X_{\leq k-1} = I_{r_{k-1}}$ , we expand the matrix  $X_{\leq k}^\top X_{\leq k}$  inductively:

$$\begin{aligned} X_{\leq k}^\top X_{\leq k} &= (U_k^L)^\top (I_{n_k} \otimes X_{\leq k-1}^\top) (I_{n_k} \otimes X_{\leq k-1}) U_k^L \\ &= (U_k^L)^\top (I_{n_k} \otimes X_{\leq k-1}^\top X_{\leq k-1}) U_k^L \\ &= (U_k^L)^\top U_k^L \\ &= I_{r_k} \end{aligned}$$

This concludes the proof for the left-orthogonal case; the right-orthogonal case can be proven in a similar manner.  $\blacksquare$

### B.4. Deriving formulas for cores from formulas of flattenings

Recall from eq. (9) that we aim to find a formula for  $\mathcal{P}_X^i(\text{D}_V \mathcal{P}_X^j)(Z)$ , which is a tensor in the  $i$ th orthogonal component of  $\text{T}_X \mathcal{M}_r$ . We can then represent  $\mathcal{P}_X^i(\text{D}_V \mathcal{P}_X^j)(Z)$  by a variational core  $\delta \tilde{V}_i$ , given by eq. (4). Using Lemma 9 and the inductive definition of  $X_{\leq k}$ , we see that eq. (4) is equivalent to the following:

$$(\mathcal{P}_X^i(\text{D}_V \mathcal{P}_X^j)(Z))^{<i>} = (I_{n_i} \otimes X_{\leq i-1}) \delta \tilde{V}_i^L \tilde{X}_{\geq i+1}^\top. \quad (10)$$

We would then hope that if we can find an equation for  $(\mathcal{P}_X^i(\text{D}_V \mathcal{P}_X^j)(Z))^{<i>}$  in the form  $(I_{n_i} \otimes X_{\leq i-1}) B \tilde{X}_{\geq i+1}^\top$ , where  $B \in \mathbb{R}^{r_{i-1} n_i \times r_i}$ , that we have uniqueness:  $B = \delta \tilde{V}_i^L$ . This is indeed the case.

**Lemma 11** *If a matrix  $B \in \mathbb{R}^{n_i r_{i-1} \times r_i}$  satisfies  $(\mathcal{P}_X^i(\mathcal{D}_V \mathcal{P}_X^j)(Z))^{\langle i \rangle} = (I_{n_i} \otimes X_{\leq i-1}) B \tilde{X}_{\geq i+1}^\top$ , then  $B = \delta \tilde{V}_i^L$ .*

**Proof** From Lemma 9,  $(\mathcal{P}_X^i(\mathcal{D}_V \mathcal{P}_X^j)(Z))^{\langle i \rangle} = (I_{n_i} \otimes X_{\leq i-1}) \delta \tilde{V}_i^L \tilde{X}_{\geq i+1}^\top$ . Supposing that a small matrix  $B$  satisfies  $(\mathcal{P}_X^i(Z))^{\langle i \rangle} = (I_{n_i} \otimes X_{\leq i-1}) B \tilde{X}_{\geq i+1}^\top$ , we then have:

$$\begin{aligned} (I_{n_i} \otimes X_{\leq i-1}) \delta \tilde{V}_i^L \tilde{X}_{\geq i+1}^\top &= (I_{n_i} \otimes X_{\leq i-1}) B \tilde{X}_{\geq i+1}^\top \\ (I_{n_i} \otimes X_{\leq i-1}^\top X_{\leq i-1}) \delta \tilde{V}_i^L \tilde{X}_{\geq i+1}^\top \tilde{X}_{\geq i+1} &= (I_{n_i} \otimes X_{\leq i-1}^\top X_{\leq i-1}) B \tilde{X}_{\geq i+1}^\top \tilde{X}_{\geq i+1} \\ \delta \tilde{V}_i^L &= B. \end{aligned}$$

■

Thus, finding a formula for  $(\mathcal{P}_X^i(\mathcal{D}_V \mathcal{P}_X^j)(Z))^{\langle i \rangle}$  in the form  $(\mathcal{P}_X^i(\mathcal{D}_V \mathcal{P}_X^j)(Z))^{\langle i \rangle} = (I_{n_i} \otimes X_{\leq i-1}) B \tilde{X}_{\geq i+1}^\top$  is sufficient to find a formula for  $\delta \tilde{V}_k$ .

### Appendix C. Proofs for alternative parametrization of $\mathbb{T}_X \mathcal{M}_r$

**Lemma 12** *For a tangent vector  $V \in \mathbb{T}_X \mathcal{M}_r$ ,  $\delta \tilde{V}_k^L = \delta V_k^L R_k^\top$ , for all  $k \in [d-1]$ , where  $R_k \in \mathbb{R}^{r_k \times r_k}$  are defined by  $X_{\geq k+1} = R_k \tilde{X}_{\geq k+1}$ , are invertible, and are generated by right-orthogonalization of  $U_1, \dots, U_d$ . Lastly,  $\delta \tilde{V}_d^L = \delta V_d^L$ .*

**Proof** Note that the last statement comes trivially from comparing definitions of the parametrizations. Furthermore, all properties of  $R_k$  come from [14, Algorithm 4.1].

Let  $V = \mathcal{P}_X(Z)$  for an arbitrary tensor  $Z$ . Note the following equality between both parametrizations using Lemma 9:

$$(\mathcal{P}_X^k(Z))^{\langle k \rangle} = (I_{n_k} \otimes X_{\leq k-1}) \delta \tilde{V}_k^L \tilde{X}_{\geq k+1}^\top = (I_{n_k} \otimes X_{\leq k-1}) \delta V_k^L X_{\geq k+1}^\top.$$

Using the relation  $X_{\geq k+1} = \tilde{X}_{\geq k+1} R_k$ , we then get that:

$$\begin{aligned} (I_{n_k} \otimes X_{\leq k-1}) \delta \tilde{V}_k^L \tilde{X}_{\geq k+1}^\top &= (I_{n_k} \otimes X_{\leq k-1}) \delta V_k^L R_{k+1}^\top \tilde{X}_{\geq k+1}^\top \\ \delta \tilde{V}_k^L &= \delta V_k^L R_{k+1}^\top \end{aligned}$$

by Lemma 13. ■

Finally, we show a uniqueness result regarding formulas for  $\delta \tilde{V}_k$  from  $(\mathcal{P}_X^k(Z))^{\langle k \rangle}$ :

**Lemma 13** *If a small matrix  $B \in \mathbb{R}^{n_i r_{i-1} \times r_i}$  satisfies  $(\mathcal{P}_X^k(Z))^{\langle k \rangle} = (I_{n_i} \otimes X_{\leq i-1}) B \tilde{X}_{\geq i+1}^\top$ , then  $B = \delta \tilde{V}_i^L$ .*

**Proof** From Lemma 9,  $(\mathcal{P}_X^k(Z))^{\langle k \rangle} = (I_{n_i} \otimes X_{\leq i-1}) \delta \tilde{V}_i^L \tilde{X}_{\geq i+1}^\top$ . Supposing that a small matrix  $B$  satisfies  $(\mathcal{P}_X^k(Z))^{\langle k \rangle} = (I_{n_i} \otimes X_{\leq i-1}) B \tilde{X}_{\geq i+1}^\top$ , we then have:

$$\begin{aligned} (I_{n_i} \otimes X_{\leq i-1}) \delta \tilde{V}_i^L \tilde{X}_{\geq i+1}^\top &= (I_{n_i} \otimes X_{\leq i-1}) B \tilde{X}_{\geq i+1}^\top \\ (I_{n_i} \otimes X_{\leq i-1}^\top X_{\leq i-1}) \delta \tilde{V}_i^L \tilde{X}_{\geq i+1}^\top \tilde{X}_{\geq i+1} &= (I_{n_i} \otimes X_{\leq i-1}^\top X_{\leq i-1}) B \tilde{X}_{\geq i+1}^\top \tilde{X}_{\geq i+1} \\ \delta \tilde{V}_i^L &= B, \end{aligned}$$

which completes the proof. ■

## Appendix D. Proofs for the correction term

In this section, we prove an explicit formula for the Riemannian Hessian of  $\mathcal{M}_r$ . We take the definition of the Riemannian Hessian to be that given in eq. (8).

### D.1. Extending $(D_V \mathcal{P}_X^i)(Z)$ to a standard differential

We start from the differential of the full orthogonal projector,  $(D_V \mathcal{P}_X)(Z)$ . Recall the definition of  $(D_V \mathcal{P}_X)(Z)$  from the main paper:

$$(D_V \mathcal{P}_X)Z = \lim_{t \rightarrow 0} \frac{\mathcal{P}_{c(t)}(Z) - \mathcal{P}_{c(0)}(Z)}{t} \quad (11)$$

where  $c(t)$  is a smooth curve on  $\mathcal{M}_r$  such that  $c(0) = X$  and  $c'(0) = V$ . Recall from eq. (5) that, given a left-orthogonal and minimal decomposition  $U_1, \dots, U_d$  of  $X$ , we can split  $\mathcal{P}_X(Z)$  into the sum  $\sum_{i=1}^d \mathcal{P}_X^i(Z)$ , where each  $\mathcal{P}_X^i(Z)$  is given by eq. (6) and (7). Thus, to be able to make this split over the curve  $c(t)$ , we want to construct  $c(t)$  as a ‘‘curve of cores’’ in the following way.

Note that each of the cores  $U_k$  of a TT-decomposition lives in the linear space  $\mathbb{R}^{r_{k-1} \times n_k \times r_k}$ . We construct a smooth map from  $(\mathbb{R}^{r_0 \times n_1 \times r_1}) \times (\mathbb{R}^{r_1 \times n_2 \times r_2}) \times \dots \times (\mathbb{R}^{r_d \times n_d \times r_{d+1}})$  to  $\mathbb{R}^{n_1 \times \dots \times n_d}$  through eq. (1) of the main paper:

$M_{TT}(W_1, W_2, \dots, W_d)$  is a tensor of size  $(n_1, \dots, n_d)$  with value  $W_1(i_1)W_2(i_2) \cdots W_d(i_d)$  at index  $(i_1, \dots, i_d)$

We denote the input space  $L_{\text{cores}}$  and the resulting map  $M_{TT}: L_{\text{cores}} \rightarrow \mathbb{R}^{n_1 \times \dots \times n_d}$ . Note that  $M_{TT}$  is surjective to  $\mathcal{M}_r$  but not injective. Then the curve  $c_{TT}(t) := M_{TT}(c_1(t), c_2(t), \dots, c_d(t))$ , where  $c_k(t)$  is a smooth curve on  $\mathbb{R}^{r_{k-1} \times n_k \times r_k}$ , is a smooth curve in  $\mathbb{R}^{n_1 \times \dots \times n_d}$ .

Finally, we can write eq. (11) as the following:

$$(D_V \mathcal{P}_X)Z = \sum_{i=1}^d \lim_{t \rightarrow 0} \frac{\mathcal{P}_{c_{TT}(t)}^i(Z) - \mathcal{P}_{c_{TT}(0)}^i(Z)}{t} \quad (12)$$

where  $c_{TT}(t)$  is defined as above,  $c_{TT}(0) = X$ ,  $c'_{TT}(0) = V$ , and for every fixed  $t$  in some small neighborhood  $(-\epsilon, \epsilon)$ ,  $c_1(t), \dots, c_d(t)$  is a left-orthogonal, minimal TT-decomposition and  $c_{TT}(t) \in \mathcal{M}_r$ . We now establish the existence of such a curve  $c_{TT}(t)$ :

**Lemma 14** *There exists a smooth curve  $c_{TT}(t)$  on  $\mathcal{M}_r$  such that  $c_{TT}(0) = X$ ,  $c'_{TT}(0) = V$ , and for every fixed  $t \in (-\epsilon, \epsilon)$ , where  $\epsilon > 0$  is fixed, we have that  $c_1(t), \dots, c_d(t)$  is a minimal and left-orthogonal TT-decomposition, where  $c_{TT}(t)$  is of the following form:*

$$c_{TT}(t) = M_{TT}(U_1 + t\delta V_1 + O(t^2), \dots, U_{d-1} + t\delta V_{d-1} + O(t^2), U_d + t\delta V_d). \quad (13)$$

Before starting the proof for Lemma 14, it is important to note that we represent  $V$  by the original parametrization  $\delta V_1, \dots, \delta V_d$ , i.e. from the formula

$$V(i_1, \dots, i_d) = \sum_{k=1}^d U_1(i_1) \cdots U_{k-1}(i_{k-1}) \delta V_k(i_k) U_{k+1}(i_{k+1}) \cdots U_d(i_d). \quad (14)$$

This is in contrast to the alternative parametrization  $\delta\tilde{V}_1, \dots, \delta\tilde{V}_d$  from which we get the formulas (6) and (7) for the orthogonal projector. We use the original parametrization because it is more intuitive to construct a curve  $c_{TT}(t)$  such that  $c'_{TT}(0) = V$  using  $\delta V_1, \dots, \delta V_d$  due to the resemblance of eq. (14) to the product rule. We now prove a note from Section C that we are able to easily interchange between these two parametrizations:

Thus, the original parametrization  $\delta V_1, \dots, \delta V_d$  is always accessible from the parametrization  $\delta\tilde{V}_1, \dots, \delta\tilde{V}_d$ . We now move on to prove Lemma 14.

**Proof** First, we will construct a curve  $c_{TT}(t)$  that satisfies all requirements except for  $c_1(t), \dots, c_d(t)$  to be minimal. Then, we will simply choose  $\epsilon$  small enough such that  $c_1(t), \dots, c_d(t)$  is also minimal for  $t \in (-\epsilon, \epsilon)$ .

We start by constructing a curve satisfying left-orthogonality. Left-orthogonality requires that  $(c_k(t))^L$  has orthonormal columns for all  $k \in [d-1]$ . Note here that we can use the structure of a different manifold, the *Stiefel manifold*:

$$\text{St}(n, p) = \left\{ M \in \mathbb{R}^{n \times p} : M^\top M = I_p \right\}$$

where the tangent space  $T_M \text{St}(n, p)$  at a point  $M$  is the set of all matrices  $A$  such that  $M^\top A + A^\top M = 0$ . More details on the Stiefel manifold can be found in [4, §7.3]. Note that for every  $k \in [d-1]$ ,  $U_k^L \in \text{St}(r_{k-1}n_k, r_k)$  by left-orthogonality. Furthermore,  $(\delta V_k^L)^\top U_k^L = 0$  (called the *gauge conditions*) by definition of the parametrization  $\delta V_1, \dots, \delta V_d$  of  $V$ , and thus  $(\delta V_k^L)^\top U_k^L + (U_k^L)^\top \delta V_k^L = 0 \implies \delta V_k^L \in T_{U_k^L} \text{St}(r_{k-1}n_k, r_k)$ . This then allows us to construct a smooth curve  $(c_k(t))^L = U_k^L + t\delta V_k^L + O(t^2)$  such that, for all  $t$  in some small neighborhood  $(-\epsilon_0, \epsilon_0)$ , we have that  $(c_k(t))^L \in \text{St}(r_{k-1}n_k, r_k)$ . Construct  $c_k(t)$  by unfolding  $(c_k(t))^L$  back into a tensor of order 3 so that  $c_k(t) = U_k + t\delta V_k + O(t^2)$ , repeat this construction for all  $k \in [d-1]$ , and finally construct  $c_d(t) = U_d + t\delta V_d$ .

It is easy to verify that the resulting curve  $c_{TT}(t) = M_{TT}(U_1 + t\delta V_1 + O(t^2), \dots, U_{d-1} + t\delta V_{d-1} + O(t^2), U_d + t\delta V_d)$  satisfies  $c_{TT}(0) = X$  and  $c'_{TT}(0) = V$ , since  $M_{TT}$  is a multilinear map. We are then left to prove that there exists a neighborhood  $(-\epsilon, \epsilon)$  small enough such that  $c_1(t), \dots, c_d(t)$  is minimal for all  $(-\epsilon, \epsilon)$ . There is a statement in [15, §9.3.3] stating that the decomposition  $c_1(t), \dots, c_d(t)$  is minimal if and only if  $(c_k(t))^L$  and  $(c_k(t))^R$  are both of full rank for all  $k \in [d]$ . This statement can be proven using Lemma 9 and rank arguments, and it is useful because the set of full rank matrices is an open set. Therefore, for each core  $U_k$ , we can construct an open ball around  $U_k$  in the space  $\mathbb{R}^{r_{k-1} \times n_k \times r_k}$  of some positive radius  $\epsilon_L^k$ , denoted  $\mathcal{N}_{\epsilon_L^k}(U_k)$ , such that all tensors in  $\mathcal{N}_{\epsilon_L^k}(U_k)$  have left-flattenings of full rank. Similarly, we can construct an open ball  $\mathcal{N}_{\epsilon_R^k}(U_k)$  for each core  $U_k$  such that all tensors in  $\mathcal{N}_{\epsilon_R^k}(U_k)$  have right-flattenings of full rank. Defining  $\epsilon_k := \min(\epsilon_L^k, \epsilon_R^k)$ , it then follows that if a TT-tensor  $W$  has cores  $W_1, \dots, W_d$  such that  $W_k \in \mathcal{N}_{\epsilon_k}(U_k)$  for all  $k \in [d]$ , then  $W_1, \dots, W_d$  is a minimal decomposition, and  $W \in \mathcal{M}_r$ . Since each  $c_k(t)$  is a smooth curve, there exists  $\delta_k > 0$  such that  $t \in (-\delta_k, \delta_k) \implies c_k(t) \in \mathcal{N}_{\epsilon_k}(U_k)$ . Setting  $\epsilon := \min(\delta_1, \dots, \delta_d, \epsilon_0)$  finishes the proof.  $\blacksquare$

Now that we have constructed  $c_{TT}$  to be a valid curve, the following lemma comes trivially:

**Lemma 15** *Let  $X \in \mathcal{M}_r$  and  $V \in T_X \mathcal{M}_r$ . The following identity holds:*

$$(\text{D}_V \mathcal{P}_X^i)Z = \lim_{t \rightarrow 0} \frac{\mathcal{P}^{i,Z}(c_{TT}(t)) - \mathcal{P}^{i,Z}(c_{TT}(0))}{t} \quad (15)$$



## D.2. Proofs for the correction term of the Riemannian Hessian

We can now find a formula for  $(D_V \mathcal{P}_X^i)Z$ . Recalling equations (6) and (7), we can evaluate  $(D_V \mathcal{P}_X^i)Z$  by evaluating the derivatives of the terms  $X_{\leq k} X_{\leq k}^\top$  and  $\tilde{X}_{> k} \tilde{X}_{> k}^\top$ . We then write  $D_V(X_{\leq k} X_{\leq k}^\top)$  to mean the following:

$$D_V(X_{\leq k} X_{\leq k}^\top) := \lim_{t \rightarrow 0} \frac{X_{\leq k}(c_{TT}(t)) X_{\leq k}^\top(c_{TT}(t)) - X_{\leq k}(c_{TT}(0)) X_{\leq k}^\top(c_{TT}(0))}{t}$$

where  $X_{\leq k}(c_{TT}(t))$  is the  $k$ th left interface matrix using cores from  $c_{TT}(t)$ . We now build up derivations for  $D_V(X_{\leq k} X_{\leq k}^\top)$  and  $D_V(X_{\geq k} X_{\geq k}^\top)$ .

**Lemma 16** *Let  $X \in \mathcal{M}_r$  and  $V \in \mathbb{T}_X \mathcal{M}_r$ . The following identities hold for all  $k \in [d]$ :*

1.  $D_V(X_{\leq k}) = V_{\leq k}$
2.  $D_V(X_{\geq k}) = V_{\geq k}$

**Proof** We will only write a proof for part 1, as the proof for part 2 is analogous.

We proceed via induction. Note that by construction of  $c_{TT}(t)$ , it follows that for all  $k \in [d]$ ,  $D_V(U_k^L) = \delta V_k^L$ . This covers the base case, since  $D_V(X_{\leq 1}) = D_V(U_1^L) = \delta V_1^L = V_{\leq 1}$ . Now, assuming the statement holds for  $k-1$ , we conclude that  $D_V(I_{n_k} \otimes X_{\leq k-1}) = I_{n_k} \otimes V_{\leq k-1}$  and that:

$$\begin{aligned} D_V(X_{\leq k}) &= D_V((I_{n_k} \otimes X_{\leq k-1})U_k^L) \\ &= D_V(I_{n_k} \otimes X_{\leq k-1})U_k^L + (I_{n_k} \otimes X_{\leq k-1})D_V(U_k^L) \\ &= (I_{n_k} \otimes V_{\leq k-1})U_k^L + (I_{n_k} \otimes X_{\leq k-1})\delta V_k^L = V_{\leq k}. \end{aligned}$$

■

Using Lemma 16, we can then evaluate  $D_V(X_{\leq k} X_{\leq k}^\top)$  through a standard application of the product rule.

**Lemma 17** *The following identity holds:*

$$D_V(X_{\leq k} X_{\leq k}^\top) = V_{\leq k} X_{\leq k}^\top + X_{\leq k} V_{\leq k}^\top \quad (16)$$

Next, we evaluate  $D_V(\tilde{X}_{> k} \tilde{X}_{> k}^\top)$ . For this differential, we need the following useful cancellation identity: Furthermore, the left variational interface matrices  $V_{\leq k}$  in particular have an important cancellation identity.

**Lemma 18** *For all  $k \in [d-1]$ , the following cancellation identity holds:  $V_{\leq k}^\top X_{\leq k} = 0$ .*

**Proof** We prove by induction. The base case  $k=1$  is given directly by the gauge conditions, since  $V_{\leq 1}^\top X_{\leq 1} = (\delta V_1^L)^\top U_1^L = 0$ . Assuming the desired cancellation holds for  $k-1$ , we expand  $V_{\leq k}^\top X_{\leq k}$  inductively via definitions to get:

$$\begin{aligned} &((I_{n_k} \otimes V_{\leq k-1})U_k^L + (I_{n_k} \otimes X_{\leq k-1})\delta V_k^L)^\top ((I_{n_k} \otimes X_{\leq k-1})U_k^L) \\ &= (U_k^L)^\top (I_{n_k} \otimes V_{\leq k-1}^\top X_{\leq k-1})U_k^L + (\delta V_k^L)^\top (I_{n_k} \otimes X_{\leq k-1}^\top X_{\leq k-1})U_k^L \\ &= 0 + (\delta V_k^L)^\top U_k^L = 0 \end{aligned}$$



■

**Lemma 19** *The following equation holds:*

$$\begin{aligned} D_V(\tilde{X}_{\geq k+1}\tilde{X}_{\geq k+1}^\top) &= (I_{n_{k+1}\cdots n_d} - \tilde{X}_{\geq k+1}\tilde{X}_{\geq k+1}^\top)V_{\geq k+1}R_k^{-1}\tilde{X}_{\geq k+1}^\top \\ &\quad + \tilde{X}_{\geq k+1}R_k^{-\top}V_{\geq k+1}^\top(I_{n_{k+1}\cdots n_d} - \tilde{X}_{\geq k+1}\tilde{X}_{\geq k+1}^\top) \end{aligned}$$

**Proof** We will first introduce an index-free shorthand notation for the sake of readability; this is used for all subsequent proofs in this paper. For example,  $R := R_k$ ,  $X_{>} := X_{>k}$ ,  $X_{\geq} := X_{\geq k}$ ,  $X_{>} := X_{>k+1}$ , and the same notation for the left interface matrices. All identity matrices are abbreviated to  $I$  with size implied from context, e.g.  $(I \otimes X_{<}) = (I_{n_k} \otimes X_{<k-1})$ .

We start the proof by differentiating both sides of the defining equality for  $R_k$ , namely  $X_{>} = \tilde{X}_{>}R$

$$\begin{aligned} D_V(X_{>}) &= D_V(\tilde{X}_{>}R) \\ V_{>} &= D_V(\tilde{X}_{>})R + \tilde{X}_{>}D_V(R) \\ V_{>} &= (\tilde{X}_{>}\Omega + \tilde{X}_{>}^\perp B)R + \tilde{X}_{>}D_V(R) \end{aligned}$$

where  $\Omega \in \mathbb{R}^{r_k \times r_k}$  is skew-symmetric, and  $B \in \mathbb{R}^{n_{k+1}\cdots n_d - r_k \times r_k}$ . This decomposition comes from the fact that  $\tilde{W}_{>}(X_t^V)$  is a smooth curve on the Stiefel manifold, so we can use the structure of its tangent space to generate  $\Omega, B$ . We want to find more explicit structure for the two unknowns we artificially introduced:  $\Omega$  and  $B$ . Starting with  $B$ , we multiply both sides by  $(\tilde{X}_{>}^\perp)^\top$ , where  $\tilde{X}_{>}^\perp$  is the orthonormal complement of  $\tilde{X}_{>}$ , on the left to get the following expression for  $B$ :

$$\begin{aligned} (\tilde{X}_{>}^\perp)^\top V_{>} &= BR \\ B &= (\tilde{X}_{>}^\perp)^\top V_{>}R^{-1} \end{aligned}$$

We then proceed to evaluate the desired equality:

$$\begin{aligned} &= (\tilde{X}_{>}\Omega + \tilde{X}_{>}^\perp B)\tilde{X}_{>}^\top + \tilde{X}_{>}(\tilde{X}_{>}\Omega + \tilde{X}_{>}^\perp B)^\top \\ &= (\tilde{X}_{>}\Omega + \tilde{X}_{>}^\perp(\tilde{X}_{>}^\perp)^\top V_{>}R^{-1})\tilde{X}_{>}^\top + \tilde{X}_{>}(\tilde{X}_{>}\Omega + \tilde{X}_{>}^\perp(\tilde{X}_{>}^\perp)^\top V_{>}R^{-1})^\top \\ &= (\tilde{X}_{>}\Omega + \tilde{X}_{>}^\perp(\tilde{X}_{>}^\perp)^\top V_{>}R^{-1})\tilde{X}_{>}^\top + \tilde{X}_{>}(\Omega^\top \tilde{X}_{>}^\top + R^{-T}V_{>}^\top \tilde{X}_{>}^\perp(\tilde{X}_{>}^\perp)^\top) \\ &= \tilde{X}_{>}^\perp(\tilde{X}_{>}^\perp)^\top V_{>}R^{-1}\tilde{X}_{>}^\top + \tilde{X}_{>}R^{-T}V_{>}^\top \tilde{X}_{>}^\perp(\tilde{X}_{>}^\perp)^\top + \tilde{X}_{>}\Omega\tilde{X}_{>}^\top + \tilde{X}_{>}\Omega^\top \tilde{X}_{>}^\top \\ &= (I - \tilde{X}_{>}\tilde{X}_{>}^\top)V_{>}R^{-1}\tilde{X}_{>}^\top + \tilde{X}_{>}R^{-T}V_{>}^\top(I - \tilde{X}_{>}\tilde{X}_{>}^\top) + \tilde{X}_{>}(\Omega + \Omega^\top)\tilde{X}_{>}^\top \\ &= (I - \tilde{X}_{>}\tilde{X}_{>}^\top)V_{>}R^{-1}\tilde{X}_{>}^\top + \tilde{X}_{>}R^{-T}V_{>}^\top(I - \tilde{X}_{>}\tilde{X}_{>}^\top) \end{aligned}$$

■

Using Lemmas 17 and 19, we can then evaluate the differential  $D_V\mathcal{P}_X^k Z$  through a simple application of the product rule:

**Lemma 20** *The differential for  $\mathcal{P}_i Z$  is given by:*

$$\begin{aligned} (\mathbf{D}_V \mathcal{P}_X^k Z)^{\langle k \rangle} &= \left( I_{n_k} \otimes (V_{<k} X_{<k}^\top + X_{<k} V_{<k}^\top) - (V_{\leq k} X_{\leq k}^\top + X_{\leq k} V_{\leq k}^\top) \right) Z^{\langle k \rangle} \tilde{X}_{>k} \tilde{X}_{>k}^\top \\ &\quad + (I_{n_k} \otimes X_{<k} X_{<k}^\top - X_{\leq k} X_{\leq k}^\top) Z^{\langle k \rangle} (I_{n_{k+1} \dots n_d} - \tilde{X}_{>k} \tilde{X}_{>k}^\top) V_{>k} R_i^{-1} \tilde{X}_{>k}^\top \\ &\quad + (I_{n_k} \otimes X_{<k} X_{<k}^\top - X_{\leq k} X_{\leq k}^\top) Z^{\langle k \rangle} \tilde{X}_{>k} R_k^{-T} V_{>k}^\top (I_{n_{k+1} \dots n_d} - \tilde{X}_{>k} \tilde{X}_{>k}^\top), \end{aligned}$$

for  $i < d$ , and

$$(\mathbf{D}_V \mathcal{P}_X^d Z)^{\langle d \rangle} = (I \otimes (V_{<d} X_{<d}^\top + X_{<d} V_{<d}^\top)) Z^{\langle d \rangle}.$$

We can then directly apply Lemma 20 to prove Theorem 2 of the main paper.

**Proof** [Proof of Theorem 2] Following the convention for the shorthand introduced above. For the  $k < d$  case:

$$\begin{aligned} &(\mathcal{P}_X^k \mathbf{D}_V \mathcal{P}_X^k Z)^{\langle k \rangle} \\ &= (\mathcal{P}_X^k \left( (I \otimes (V_{<k} X_{<k}^\top + X_{<k} V_{<k}^\top) - (V_{\leq k} X_{\leq k}^\top + X_{\leq k} V_{\leq k}^\top)) Z^{\langle k \rangle} \tilde{X}_{>k} \tilde{X}_{>k}^\top \right))^{\langle k \rangle} \quad (17) \end{aligned}$$

$$+ (\mathcal{P}_X^k \left( (I \otimes X_{<k} X_{<k}^\top - X_{\leq k} X_{\leq k}^\top) Z^{\langle k \rangle} (I - \tilde{X}_{>k} \tilde{X}_{>k}^\top) V_{>k} R_i^{-1} \tilde{X}_{>k}^\top \right))^{\langle k \rangle} \quad (18)$$

$$+ (\mathcal{P}_X^k \left( (I \otimes X_{<k} X_{<k}^\top - X_{\leq k} X_{\leq k}^\top) Z^{\langle k \rangle} \tilde{X}_{>k} R^{-T} V_{>k}^\top (I - \tilde{X}_{>k} \tilde{X}_{>k}^\top) \right))^{\langle k \rangle}. \quad (19)$$

Recall the formula for  $(\mathcal{P}_X^k Z)^{\langle k \rangle}$  given in eq. (6). The operation consists of left multiplication by  $(I_{n_k} \otimes (X_{<k} X_{<k}^\top) - X_{\leq k} X_{\leq k}^\top)$  and right multiplication by  $\tilde{X}_{>k} \tilde{X}_{>k}^\top$ . Since  $(I - \tilde{X}_{>k} \tilde{X}_{>k}^\top) \tilde{X}_{>k} \tilde{X}_{>k}^\top = \tilde{X}_{>k} \tilde{X}_{>k}^\top - \tilde{X}_{>k} \tilde{X}_{>k}^\top = 0$ , the 3rd summand (19) vanishes. Since  $(I \otimes X_{<k} X_{<k}^\top - X_{\leq k} X_{\leq k}^\top)^2 = (I \otimes X_{<k} X_{<k}^\top - X_{\leq k} X_{\leq k}^\top)$ , (18) remains unchanged from  $\mathcal{P}_X^k$ . We can simplify (17) in the following way:

$$\begin{aligned} &(\mathcal{P}_X^k \left( (I \otimes (V_{<k} X_{<k}^\top + X_{<k} V_{<k}^\top) - (V_{\leq k} X_{\leq k}^\top + X_{\leq k} V_{\leq k}^\top)) Z^{\langle k \rangle} \tilde{X}_{>k} \tilde{X}_{>k}^\top \right))^{\langle k \rangle} \\ &= (I \otimes X_{<k} X_{<k}^\top - X_{\leq k} X_{\leq k}^\top) (I \otimes (V_{<k} X_{<k}^\top + X_{<k} V_{<k}^\top) - (V_{\leq k} X_{\leq k}^\top + X_{\leq k} V_{\leq k}^\top)) Z^{\langle k \rangle} \tilde{X}_{>k} \tilde{X}_{>k}^\top \\ &= \left( I \otimes X_{<k} V_{<k}^\top - (I \otimes X_{<k}) U_k^L V_{\leq k}^\top - (I \otimes X_{<k}) \delta V_k^L X_{\leq k}^\top - X_{\leq k} (U_k^L)^\top (I \otimes V_{<k}^\top) + X_{\leq k} V_{\leq k}^\top \right) \\ &\quad \cdot Z^{\langle k \rangle} \tilde{X}_{>k} \tilde{X}_{>k}^\top \\ &= \left( I \otimes X_{<k} V_{<k}^\top - X_{\leq k} V_{\leq k}^\top - (I \otimes X_{<k}) \delta V_k^L X_{\leq k}^\top - X_{\leq k} (U_k^L)^\top (I \otimes V_{<k}^\top) + X_{\leq k} V_{\leq k}^\top \right) Z^{\langle k \rangle} \tilde{X}_{>k} \tilde{X}_{>k}^\top \\ &= \left( I \otimes X_{<k} V_{<k}^\top - (I \otimes X_{<k}) \delta V_k^L (U_k^L)^\top (I \otimes X_{<k}^\top) - (I \otimes X_{<k}) U_k^L (U_k^L)^\top (I \otimes V_{<k}^\top) \right) Z^{\langle k \rangle} \tilde{X}_{>k} \tilde{X}_{>k}^\top. \end{aligned}$$

Factoring out  $(I \otimes X_{<k})$  from (18) and adding it to the result above gets the desired result:

$$\begin{aligned} &= (I \otimes X_{<k}) \left( ((I - U_k^L (U_k^L)^\top) (I \otimes V_{<k}^\top)) Z^{\langle k \rangle} \tilde{X}_{>k} \right) \tilde{X}_{>k}^\top \\ &\quad - (I \otimes X_{<k}) \left( \delta V_k^L (U_k^L)^\top (I \otimes X_{<k}) \right) Z^{\langle k \rangle} \tilde{X}_{>k} \right) \tilde{X}_{>k}^\top \\ &\quad + (I \otimes X_{<k}) \left( (I - U_k^L (U_k^L)^\top) (I \otimes X_{<k}^\top) Z^{\langle k \rangle} (I - \tilde{X}_{>k} \tilde{X}_{>k}^\top) V_{>k} R^{-1} \right) \tilde{X}_{>k}^\top. \end{aligned}$$

The  $k = d$  case follows much more straightforwardly:

$$\begin{aligned}
 (\mathcal{P}_X^d D_V \mathcal{P}_X^d Z)^{\langle d \rangle} &= \mathcal{P}_X^d (I \otimes (V_{\langle d} X_{\langle d}^\top + X_{\langle d} V_{\langle d}^\top)) Z^{\langle d \rangle} \\
 &= (I \otimes (X_{\langle d} X_{\langle d}^\top)) (I \otimes (V_{\langle d} X_{\langle d}^\top + X_{\langle d} V_{\langle d}^\top)) Z^{\langle d \rangle} \\
 &= (I \otimes (X_{\langle d} V_{\langle d}^\top)) Z^{\langle d \rangle} \\
 &= (I \otimes X_{\langle d}) (I \otimes V_{\langle d}^\top) Z^{\langle d \rangle}
 \end{aligned}$$

■

Recall that in the main paper we introduced the formula  $\mathcal{P}_X^i D_V \mathcal{P}_X^j Z = -(D_V \mathcal{P}_X^i) \mathcal{P}_X^j Z$  as a key component for deriving the cross-term formulae. We provide here a proof of this statement:

**Lemma 21** *For all  $i, j \in [d], i \neq j$ , the following identity holds:  $\mathcal{P}_X^i D_V \mathcal{P}_X^j Z = -(D_V \mathcal{P}_X^i) \mathcal{P}_X^j Z$*

**Proof** First, note that  $\mathcal{P}_X^i \mathcal{P}_X^j Z = 0$  for all  $X \in \mathcal{M}_r$  by orthogonality. From this, we directly get that  $D_V(\mathcal{P}_X^i \mathcal{P}_X^j Z) = 0$ . From Lemma 20, we know that differentials of the form  $D_V \mathcal{P}_X^k Z$  exist for all fixed tensors  $Z$ , allowing us to split  $D_V(\mathcal{P}_X^i \mathcal{P}_X^j Z)$  by the product rule:

$$\begin{aligned}
 D_V(\mathcal{P}_X^i \mathcal{P}_X^j Z) &= 0 \\
 \lim_{t \rightarrow 0} \frac{\mathcal{P}_{c(t)}^i \mathcal{P}_{c(t)}^j Z - \mathcal{P}_{c(0)}^i \mathcal{P}_{c(0)}^j Z}{t} &= 0 \\
 \lim_{t \rightarrow 0} \frac{\mathcal{P}_{c(t)}^i \mathcal{P}_{c(t)}^j Z - \mathcal{P}_{c(0)}^i \mathcal{P}_{c(t)}^j Z + \mathcal{P}_{c(0)}^i \mathcal{P}_{c(t)}^j Z - \mathcal{P}_{c(0)}^i \mathcal{P}_{c(0)}^j Z}{t} &= 0 \\
 \lim_{t \rightarrow 0} \frac{(\mathcal{P}_{c(t)}^i - \mathcal{P}_{c(0)}^i) \mathcal{P}_{c(t)}^j Z}{t} + \lim_{t \rightarrow 0} \frac{\mathcal{P}_{c(0)}^i (\mathcal{P}_{c(t)}^j Z - \mathcal{P}_{c(0)}^j Z)}{t} &= 0 \\
 (D_V \mathcal{P}_X^i) \mathcal{P}_X^j Z + \mathcal{P}_X^i D_V \mathcal{P}_X^j Z &= 0 \\
 \mathcal{P}_X^i D_V \mathcal{P}_X^j Z &= -(D_V \mathcal{P}_X^i) \mathcal{P}_X^j Z
 \end{aligned}$$

■

Finally, we will prove Theorem 3 of the main paper:

**Proof** [Proof for Theorem 3] Denote  $Y_{\leq}^j$  and  $Y_{>}^j$  to be interface matrices for a TT-decomposition  $U_1, \dots, U_{j-1}, \delta Y_j, \tilde{U}_{j+1}, \dots, \tilde{U}_d$ , where  $\delta Y_j$  are the variational cores for the tangent vector  $\mathcal{P}_X^j(Z)$ . Starting with the  $j > i$  case, we can use Theorem 3.6 to write the following:

$$\begin{aligned}
 (\mathcal{P}_i D_V \mathcal{P}_j Z)^{\langle i \rangle} &= -(D_V \mathcal{P}_i) \mathcal{P}_j Z^{\langle i \rangle} \\
 &= - \left( I \otimes (V_{\langle i} X_{\langle i}^\top + X_{\langle i} V_{\langle i}^\top) - (V_{\leq i} X_{\leq i}^\top + X_{\leq i} V_{\leq i}^\top) \right) X_{\leq i} (Y_{>}^j)^\top \tilde{X}_{> i} \tilde{X}_{> i}^\top \\
 &\quad - (I \otimes X_{\langle i} X_{\langle i}^\top - X_{\leq i} X_{\leq i}^\top) X_{\leq i} (Y_{>}^j)^\top (I - \tilde{X}_{>} \tilde{X}_{>}^\top) V_{>} R^{-1} \tilde{X}_{>}^\top \\
 &\quad - (I \otimes X_{\langle i} X_{\langle i}^\top - X_{\leq i} X_{\leq i}^\top) X_{\leq i} (Y_{>}^j)^\top \tilde{X}_{>} R^{-T} V_{>}^\top (I - \tilde{X}_{>} \tilde{X}_{>}^\top)
 \end{aligned}$$

Along with the identity  $V_{\leq k}^\top X_{\leq k} = 0$ , it holds that  $(I \otimes X_{<} X_{<}^\top - X_{\leq} X_{\leq}^\top) X_{\leq} = 0$ :

$$\begin{aligned}
 (I \otimes X_{<} X_{<}^\top - X_{\leq} X_{\leq}^\top) X_{\leq} &= (I \otimes X_{<} X_{<}^\top) X_{\leq} - X_{\leq} X_{\leq}^\top X_{\leq} \\
 &= (I \otimes X_{<} X_{<}^\top) (I \otimes X_{<}) U^L - X_{\leq} \\
 &= (I \otimes X_{<} X_{<}^\top X_{<}) U^L - X_{\leq} \\
 &= (I \otimes X_{<}) U^L - X_{\leq} \\
 &= X_{\leq} - X_{\leq} = 0
 \end{aligned}$$

Thus, we are left only with the first summand:

$$\begin{aligned}
 -(D_V \mathcal{P}_i) \mathcal{P}_j Z)^{<i>} &= -(I \otimes (V_{<} X_{<}^\top) - V_{\leq} X_{\leq}^\top) X_{\leq} (Y_{>}^j)^\top \tilde{X}_{>} \tilde{X}_{>}^\top \\
 &= (V_{\leq} - (I \otimes V_{<}) U^L) (Y_{>}^j)^\top \tilde{X}_{>} \tilde{X}_{>}^\top \\
 &= (I \otimes X_{<}) \delta U_i^L (Y_{>}^j)^\top \tilde{X}_{>} \tilde{X}_{>}^\top
 \end{aligned}$$

Giving us the desired form for the  $j > i$  case. For the  $j < i$  case, we have the following similar expression:

$$\begin{aligned}
 -(D_V \mathcal{P}_i) \mathcal{P}_j Z)^{<i>} &= - \left( I \otimes (V_{<} X_{<}^\top + X_{<} V_{<}^\top) - (V_{\leq} X_{\leq}^\top + X_{\leq} V_{\leq}^\top) \right) Y_{\leq}^j \tilde{X}_{>}^\top \tilde{X}_{>} \tilde{X}_{>}^\top \\
 &\quad - (I \otimes X_{<} X_{<}^\top - X_{\leq} X_{\leq}^\top) Y_{\leq}^j \tilde{X}_{>}^\top (I - \tilde{X}_{>} \tilde{X}_{>}^\top) V_{>} R^{-1} \tilde{X}_{>}^\top \\
 &\quad - (I \otimes X_{<} X_{<}^\top - X_{\leq} X_{\leq}^\top) Y_{\leq}^j \tilde{X}_{>}^\top \tilde{X}_{>} R^{-T} V_{>}^\top (I - \tilde{X}_{>} \tilde{X}_{>}^\top)
 \end{aligned}$$

Note that both  $X_{\leq k}^\top Y_{\leq k}^j = 0$  and  $X_{< k}^\top Y_{< k}^j = 0$ ; this is because  $Y_{\leq k}^j$  and  $Y_{< k}^j$  are simply variational interface matrices for a tangent vector with cores  $\{0, \dots, 0, \delta Y_j, 0, \dots, 0\}$ , allowing us to use Lemma 18. It follows that both  $X_{\leq}^\top Y_{\leq}^j = 0$  and  $(I \otimes X_{<}^\top) Y_{\leq}^j = (I \otimes X_{<}^\top Y_{<}^j) U^L = 0$ , so the second and third summands are indeed 0. We are then left with the first summand:

$$\begin{aligned}
 -(D_V \mathcal{P}_i) \mathcal{P}_j Z)^{<i>} &= - \left( I \otimes (X_{<} V_{<}^\top) - (X_{\leq} V_{\leq}^\top) \right) Y_{\leq}^j \tilde{X}_{>}^\top \tilde{X}_{>} \tilde{X}_{>}^\top \\
 &= -(I \otimes X_{<} V_{<}^\top - X_{\leq} V_{\leq}^\top) Y_{\leq}^j \tilde{X}_{>}^\top
 \end{aligned}$$

This finishes the  $j < i$  case. The 3rd case,  $(\mathcal{P}_X^d D_V \mathcal{P}_X^j Z)^{<d>}$ , is trivial. ■

## Appendix E. Computational complexity analysis

In this section, we will outline an algorithm to compute the correction term on  $\mathcal{M}_r$  given formulas from Theorems 2 and 3. In general, we want to avoid the ‘‘curse of dimensionality’’ by avoiding any computation on the order of  $O(n^d)$ . For simplicity, we restrict the rank and size vectors in this section to be uniform, i.e.  $\mathbf{n} = (n, \dots, n)$  and  $\mathbf{r} = (r, \dots, r)$ .

To begin this section, we establish a number of computations that we know can be done in feasible time:

**Corollary 22** *The following can be computed in  $O(dnr^3)$  flops:*

1. *Given a TT-tensor  $X$  with left-orthogonal decomposition  $\{U_1, \dots, U_d\}$ , the right-orthogonal decomposition  $\{\tilde{U}_1, \dots, \tilde{U}_d\}$  along with matrices  $\{R_1, \dots, R_{d-1}\}$  as defined in Section B.2 are computable in  $O(dnr^3)$  flops.*
2. *Given two TT-tensors  $X$  and  $Y$  of size  $\mathbf{n}$ , along with their TT-decompositions of size  $\mathbf{r}$ , the set of matrices  $\{X_{\leq 1}^\top Y_{\leq 1}, \dots, X_{\leq d}^\top Y_{\leq d}\}$  is computable in  $O(dnr^3)$  flops. Similarly, the set of matrices  $\{X_{\geq 1}^\top Y_{\geq 1}, \dots, X_{\geq d}^\top Y_{\geq d}\}$  is also computable in  $O(dnr^3)$  flops.*

This is a corollary from [14, Alg. 4.1] and [14, Alg. 4.2] respectively. Note that [14, Alg. 4.2] is presented as an algorithm to compute the inner product; however, the algorithm achieves this by computing  $X_{\leq d}^\top Y_{\leq d} = \langle X, Y \rangle$ , and the intermediate steps for computing  $X_{\leq d}^\top Y_{\leq d}$  are indeed  $\{X_{\leq 1}^\top Y_{\leq 1}, \dots, X_{\leq d-1}^\top Y_{\leq d-1}\}$ . There is analogous procedure stated in [14, §4.2.3] that computes  $X_{\geq 1}^\top Y_{\geq 1}$ .

### E.1. Computational complexity for the diagonal terms

We will first show the general  $O(dz + dnr^3)$  estimate, then show the  $O(z + dnr^3)$  estimate for tensor completion and argue why we would see this estimate for typical applications. We assume that we are given a left-orthogonal decomposition  $\{U_1, \dots, U_d\}$  for TT-tensor  $X$ , as well as tangent cores  $\{\delta\tilde{V}_1, \dots, \delta\tilde{V}_d\}$  for tangent vector  $V$ .

For all cases, the limiting computations will be to compute the following:

$$\{(I \otimes X_{<k})Z^{<k>} \tilde{X}_{>k}\}_{k=1}^d, \quad (20)$$

$$\{(I \otimes V_{<k})Z^{<k>} X_{>k}\}_{k=1}^d, \text{ and} \quad (21)$$

$$\{(I \otimes X_{<k})Z^{<k>} V_{>k}\}_{k=1}^d. \quad (22)$$

After computing these matrices, we can build up the computation for the core of the diagonal term  $\mathcal{P}_X^k D_V \mathcal{P}_X^k Z$  through small matrix multiplications ( $U_k^L, \delta V_k^L \in \mathbb{R}^{rn \times r}$ ,  $R_k \in \mathbb{R}^{r \times r}$ ). Obtaining all of the  $R_k$  matrices can be done in  $O(dnr^3)$  flops by Corollary 22.2, and since  $R_k$  is upper triangular, each  $R_k^{-1}$  is computable in under  $O(nr^3)$  flops. Note we need to extract the  $\delta V_k$  parametrization from the  $\delta\tilde{V}_k$  parametrization; this can be done under  $O(dnr^3)$  flops total by Lemma 12. Finally, we are left with the computation of  $\tilde{X}_{>k}^\top V_{>k}$ ; since each matrix is in  $\mathbb{R}^{n_{k+1} \cdots n_d \times r_k}$ , computing the products explicitly would yield a curse of dimensionality. Note that this product is similar to the one given in Corollary 22.2, which is an order  $O(dnr^3)$  computation. Indeed, we are able to make a similar algorithm to compute all such  $\tilde{X}_{>k}^\top V_{>k}$  products in a total of  $O(dnr^3)$  flops. After finishing small matrix multiplications, this completes the computation of the variational cores for all diagonal terms  $\mathcal{P}_X^k D_V \mathcal{P}_X^k Z$  in an extra  $O(dnr^3)$  flops, on top of the computation of (20)(21)(22). This then leaves us to estimate the computation for (20)(21)(22).

In typical algorithms, there is some structure on the Euclidean gradient  $Z$  that allows for more efficient computation of (20)(21)(22). For example, in tensor completion, the gradient  $Z$  is sparse, and we can compute (20) in  $O(d|\Omega|r^2)$  flops using [14, Alg. 5.2], where  $\Omega$  is the set of observed indices.

As can be seen from the tensor completion example, tight estimates for the computational complexity of (20)(21)(22) will be dependent on the gradient's structure. Nonetheless, we argue that

---

**Algorithm 1** Computing  $\{\tilde{X}_{>k}^\top V_{>k}\}_{k=1}^d$  (adapted from [14, Alg. 4.2])

---

**Result:** The set of matrices  $\{\tilde{X}_{>1}^\top V_{>1}, \dots, \tilde{X}_{>d-1}^\top V_{>d-1}\}$

**Input:** TT-cores  $\{U_1, \dots, U_d\}$ ,  $\{\tilde{U}_1, \dots, \tilde{U}_d\}$ , and variational cores  $\{\delta V_1, \dots, \delta V_d\}$

$$p_1 \leftarrow \tilde{U}_d^R (\delta V_d^R)^\top$$

$$p_2 \leftarrow \tilde{U}_d^R (U_d^R)^\top$$

$$\tilde{X}_{>d-1}^\top V_{>d-1} \leftarrow p_1$$

**for**  $k = d - 1, \dots, 2$  **do**

$$A \leftarrow \tilde{U}_k \times_3 p_1^\top \text{ (} A \text{ is a tensor of size } (r_{k-1}, n_k, r_k)\text{)}$$

$$B \leftarrow \tilde{U}_k \times_3 p_2^\top$$

$$p_1 \leftarrow A^R (U_k^R)^\top + B^R (\delta V_k^R)^\top \text{ (note the resemblance of Definition 7)}$$

$$\tilde{X}_{>k-1}^\top V_{>k-1} \leftarrow p_1$$

$$p_2 \leftarrow B^R (U_k^R)^\top$$

**end**

---

in typical algorithms, computing (20)(21)(22) will typically take about as much time as computing (20), and computing the diagonal term sum  $\sum_{k=1}^d \mathcal{P}_X^k D_V \mathcal{P}_X^k(Z)$  will take about as much time as computing  $\mathcal{P}_X(Z)$ .

To illustrate this claim, we present pseudocode for an algorithm for an algorithm for efficient computing (20)(21)(22) when the gradient  $Z$  is sparse. This algorithm adapts heavily from [14, Alg. 5.2], the efficient algorithm for computing (20) for sparse gradients. This algorithm highlights the general approach to transform algorithms that compute (20) into those that compute (20)(21)(22): alongside the original computations of matrices  $(I \otimes X_{<k})Z^{<k>}\tilde{X}_{>k}$  that utilize Def. 6, we can build up  $(I \otimes V_{<k})Z^{<k>}\tilde{X}_{>k}$  and  $(I \otimes X_{<k})Z^{<k>}V_{>k}$  in the same manner that instead utilize Def. 7.

## E.2. Computational complexity for the cross terms

After establishing Corollary 22, proving the computational complexity for the cross-terms becomes very straight-forward. A fundamental object for the cross terms are the interface matrices for  $\mathcal{P}_X^j(Z)$ , denoted by  $Y_{<i}^j$  and  $Y_{>i}^j$ . Note that the variational core for  $\mathcal{P}_X^j(Z)$  is computable from (20) by left multiplication of  $(I - U_j^L (U_j^L)^\top)$ , so, given (20) has been computed, we can compute all  $Y_{<i}^j, Y_{>i}^j$  in  $O(dnr^3)$  flops.

For the  $j > i, i < d$  case, we can fix  $j$  and compute  $(Y_{<i}^j)^\top \tilde{X}_{>i}$  for all  $i < d$  in  $O(dnr^3)$  flops by Corollary 22.2. We then compute all such matrix products in a total of  $O(d^2nr^3)$  flops, and finally the small matrix multiplication of  $\delta V_i^L$  is done in  $O(nr^3)$  flops each, leading to total complexity of  $O(d^2nr^3)$  flops.

The  $j > i, i < d$  case follows similarly, where we compute all  $(I \otimes V_{<i}^\top)Y_{\leq i}^j$  for fixed  $j$  in  $O(dnr^3)$  flops using an algorithm akin to Algorithm E.1. This leads to a total  $O(d^2nr^3)$  flops, and finally the remaining  $(I - U_k^L (U_k^L)^\top)$  multiplications is done in  $O(d^2nr^3)$  flops, leading to total complexity of  $O(d^2nr^3)$  flops.

The final case is merely a computation of  $(I \otimes V_{<d}^\top)Y_{\leq d}^j$ , which we have already established can be done in  $O(dnr^3)$  flops, leading to total complexity of the entire computation in  $O(d^2nr^3)$  flops.

---

**Algorithm 2** Computation of (20)(21)(22) for Tensor Completion (adapted from [14, Alg. 5.2])

**Result:** Three set of  $d$  matrices:  $\{(I \otimes X_{<k})Z^{<k>\tilde{X}_{>k}}\}_{k=1}^d$ ,  $\{(I \otimes V_{<k})Z^{<k>X_{>k}}\}_{k=1}^d$ , and  $\{(I \otimes X_{<k})Z^{<k>V_{>k}}\}_{k=1}^d$ .

**Input:** TT-cores  $\{U_1, \dots, U_d\}$ ,  $\{\tilde{U}_1, \dots, \tilde{U}_d\}$ , variational cores  $\{\delta V_1, \dots, \delta V_d\}$ , Euclidean gradient  $Z$ , observation set  $\Omega$ 
**for**  $k = 1, \dots, d$  **do**
 $A_k \leftarrow \text{zeros}(r_{k-1}, n_k, r_k)$  (cores for computing  $(I \otimes X_{<})Z^{<k>\tilde{X}_{>}}$ )

 $B_k \leftarrow \text{zeros}(r_{k-1}, n_k, r_k)$  (cores for computing  $(I \otimes V_{<})Z^{<k>\tilde{X}_{>}}$ )

 $C_k \leftarrow \text{zeros}(r_{k-1}, n_k, r_k)$  (cores for computing  $(I \otimes X_{<})Z^{<k>V_{>}}$ )

**end**
**for**  $(i_1, \dots, i_d) \in \Omega$  **do**

 (Precompute left matrix products for  $X_{<}$  and  $V_{<}$ )

 $U_L\{1\} \leftarrow U_1(i_1)$  ( $U_L\{\}$  and  $\delta V_L\{\}$  are helper variables separate from input cores)

 $\delta V_L\{1\} \leftarrow \delta V_1(i_1)$ 
**for**  $k = 2, \dots, d-1$  **do**
 $\delta V_L\{k\} \leftarrow \delta V_L\{k-1\}U_k(i_k) + U_L\{k-1\}\delta V_k(i_k)$ 
 $U_L\{k\} \leftarrow U_L\{k-1\}U_k(i_k)$ 
**end**
 $(\tilde{U}_R, U_R, \delta V_R)$  are helper variables)

 $\tilde{U}_R \leftarrow \tilde{U}_d(i_d)$ 
 $U_R \leftarrow U_d(i_d)$ 
 $\delta V_R \leftarrow \delta V_d(i_d)$ 

(Calculate the cores beginning from the right)

 $A_d(i_d) \leftarrow A_d(i_d) + Z(i_1, \dots, i_d) \cdot U_L\{d-1\}^\top$ 
 $B_d(i_d) \leftarrow B_d(i_d) + Z(i_1, \dots, i_d) \cdot \delta V_L\{d-1\}^\top$ 
 $C_d(i_d) \leftarrow C_d(i_d) + Z(i_1, \dots, i_d) \cdot U_L\{d-1\}^\top$ 
**for**  $k = d-1, \dots, 2$  **do**
 $A_k(i_k) \leftarrow A_k(i_k) + Z(i_1, \dots, i_d) \cdot U_L\{k-1\}^\top \tilde{U}_R^\top$ 
 $B_k(i_k) \leftarrow B_k(i_k) + Z(i_1, \dots, i_d) \cdot \delta V_L\{k-1\}^\top \tilde{U}_R^\top$ 
 $C_k(i_k) \leftarrow C_k(i_k) + Z(i_1, \dots, i_d) \cdot U_L\{k-1\}^\top \delta V_R^\top$ 
 $\tilde{U}_R \leftarrow \tilde{U}_k(i_k) \tilde{U}_R^\top$ 
 $\delta V_R \leftarrow U_k(i_k) \delta V_R^\top + \delta V_k(i_k) U_R^\top$ 
 $U_R \leftarrow U_k(i_k) U_R^\top$ 
**end**
 $A_1(i_1) \leftarrow A_1(i_1) + Z(i_1, \dots, i_d) \cdot \tilde{U}_R^\top$ 
 $(V_{<1} = 0, \text{ so no update for } B \text{ here})$ 
 $C_1(i_1) \leftarrow C_1(i_1) + Z(i_1, \dots, i_d) \cdot U_L\{k-1\}^\top \delta V_R^\top$ 
**end**
**for**  $k = 1, \dots, d$  **do**
 $(I \otimes X_{<k})Z^{<k>\tilde{X}_{>k}} \leftarrow A_k^L$ 
 $(I \otimes V_{<k})Z^{<k>\tilde{X}_{>k}} \leftarrow B_k^L$ 
 $(I \otimes X_{<k})Z^{<k>V_{>k}} \leftarrow C_k^L$ 
**end**


---