

On a Multi-Stage Tensor Reduction Strategy for Arbitrary Order- p Tensorial Data under the Tensor T-Product Algebra

Harshit Kapadia, Lihong Feng, Peter Benner

Abstract

We present a novel multi-stage tensor reduction (MSTR) framework for tensorial data arising from experimental measurements or high-fidelity simulations of physical systems. The order p of the tensor under consideration can be arbitrarily large. At the heart of the framework are a series of strategic tensor factorizations and compressions, ultimately leading to a *final* order-preserving reduced representation of the original tensor. We also augment the MSTR framework by performing efficient kernel-based interpolation/regression over certain reduced tensor representations, amounting to a new non-intrusive model reduction approach capable of handling dynamical, parametric steady, and parametric dynamical systems. Furthermore, to efficiently build the parametric reduced-order model in the offline stage, we develop a tensor empirical interpolation method (t-EIM). We formalize our ideas using the tensor t-product algebra [7, 3, 6] and provide a rigorous upper bound for the error of the tensor approximation from the MSTR strategy.

The idea to factor any order-3 tensor in two orthogonal tensors of order-3 and an f -diagonal tensor of order-3 first appeared in [7]. The notion of orthogonal tensors and such a factorization strategy, analogous to matrix factorization—rendered by the singular value decomposition (SVD)—is possible due to the tensor multiplication, referred to as the t-product [7]. An extension for order- p tensors of the t-product and t-SVD is proposed in [8], which is used in our work. Moreover, by following the approach taken in [9] for order-3 tensors, we develop a randomized variant of t-SVD for order- p tensors, which is utilized to accelerate the tensor factorizations in our MSTR framework.

To aid our discussion, let us define the t-SVD for an order- p tensor $\mathcal{T}$:

$$\mathcal{T} = \mathcal{L} * \mathcal{M} * \mathcal{R}^\top, \quad (1)$$

where $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times \dots \times n_p}$, $\mathcal{L} \in \mathbb{R}^{n_1 \times n_1 \times n_3 \times \dots \times n_p}$, $\mathcal{R} \in \mathbb{R}^{n_2 \times n_2 \times n_3 \times \dots \times n_p}$, and $\mathcal{M} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times \dots \times n_p}$. Here, $\mathcal{L}$ and $\mathcal{R}$ are orthogonal, and $\mathcal{M}$ has entries $\mathcal{M}_{i_1 i_2 i_3 \dots i_p}$ such that $\mathcal{M}_{i_1 i_2 i_3 \dots i_p} = 0$ unless $i_1 = i_2$. When $p = 3$, authors in [7] refer to $\mathcal{M}$ as an f -diagonal tensor. The symbol $*$ in (1) refers to the t-product, and $\mathcal{R}^\top$ is the t-transpose of $\mathcal{R}$.

We are concerned with data arising from high-fidelity simulations or physical measurements. In the most general setting, the solution tensor $\mathcal{S}$ can have the following form:

$$\mathcal{S} \in \mathbb{R}^{N_x \times N_y \times N_z \times m_1 \times m_2 \times \dots \times m_{N_\mu} \times N_t}, \quad (2)$$

where N_x , N_y , and N_z refer to the size of each spatial dimension; N_μ refers to the parameter space dimensions, with $m_1, m_2, \dots, m_{N_\mu}$ corresponding to the size of each dimension of the parameter space; N_t refers to the total number of time steps or the frequency of measurements. We seek to efficiently reduce this high-dimensional tensorial data, obtaining its reduced tensor representation, which describes the original tensor with reasonable accuracy.

The MSTR strategy begins by identifying the *target variable* of interest, along which we do not seek to perform a tensor compression. For physical systems, it is typical to either collect measurements or high-fidelity solution values across the spatial domain, corresponding to various time instances and/or parameter configurations. As a result, the *target variable* could be either time or parameter.

We seek to compress the solution tensor along all remaining dimensions. For instance, consider an order-5 tensor $\mathcal{S} \in \mathbb{R}^{N_x \times N_y \times N_z \times m \times N_t}$, where $m = \prod_{j=\{1,2,\dots,N_\mu\}} m_j$. If the *target variable* is the parameter, then the reduced representation we aim for lies in $\mathbb{R}^{r_1 \times r_2 \times r_3 \times m \times r_5}$, whereas if the *target variable* is time, then the reduced representation we aim for lies in $\mathbb{R}^{r_1 \times r_2 \times r_3 \times r_4 \times N_t}$. Here, $r_1 \ll N_x$, $r_2 \ll N_y$, $r_3 \ll N_z$, $r_4 \ll m$, and $r_5 \ll N_t$.

Based on the t-SVD in (1), we can seek a compression of any tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times \dots \times n_p}$ by truncating $\mathcal{L} \in \mathbb{R}^{n_1 \times n_1 \times n_3 \times \dots \times n_p}$ along the second dimension, obtaining $\tilde{\mathcal{L}} \in \mathbb{R}^{n_1 \times r \times n_3 \times \dots \times n_p}$, where $r \ll n_1$, projecting $\mathcal{T}$ on $\tilde{\mathcal{L}}$, and producing $\mathcal{A} \in \mathbb{R}^{r \times n_2 \times n_3 \times \dots \times n_p}$. Note that $\mathcal{A}$ provides a reduced representation of $\mathcal{T}$, where information along the first dimension is compressed.

The central idea pertaining to the MSTR strategy is to recursively perform a tensor factorization for obtaining a truncated orthogonal tensor, onto which the parent unfactored tensor can be projected, leading to compression along one tensor dimension at every stage. The tensor factorizations are performed sequentially over subsequent *intermediate* reduced representations of the original tensor $\mathcal{S}$, ultimately leading to the *final* reduced tensor representation where all dimensions except the one corresponding to the *target variable* are compressed. While employing t-SVD to undertake the multi-stage tensor factorizations, it is imperative to appropriately permute the dimensions of the *intermediate* reduced tensor representations, allowing us to *attack* all relevant dimensions, leading to a compression of information along them. Moreover, for $\mathcal{S}$ and all subsequent *intermediate* reduced tensor representations, it is important to maintain a specific tensor orientation. We will provide further details about these intricacies in our talk.

We demonstrate an application of the MSTR strategy in the context of reduced-order modeling by using it to extract the *final* reduced tensor representation $\mathcal{A}_{n_s}$ for any given $\mathcal{S}$, along with the truncated orthogonal tensors $\{\tilde{\mathcal{L}}_i\}_{i=1}^{n_s}$ from n_s tensor reduction stages. The primary motivation is to utilize the order-preserving compressed version of $\mathcal{S}$, enabling efficient operations within our reduced-order model, which can then deliver reliable predictions during the online phase at previously unseen parameter and/or time locations. After carrying out the MSTR procedure, we interpolate/regress between specific slices of $\mathcal{A}_{n_s}$, generating a map $\mathcal{M}_{\text{tv}}$ capable of accurately rendering the *final* reduced tensor representation corresponding to new locations of the *target variable*, i.e., either parameter or time. We denote this approximation as $\hat{\mathcal{A}}_{n_s}$, which is obtained in the online phase. Note that the subscript tv in $\mathcal{M}_{\text{tv}}$ refers to the *target variable*. Using the truncated orthogonal tensors $\{\tilde{\mathcal{L}}_i\}_{i=1}^{n_s}$ and $\hat{\mathcal{A}}_{n_s}$, we obtain an approximation of the solution tensor at new locations of the *target variable*. $\mathcal{M}_{\text{tv}}$ is built using a kernel-based shallow neural network (KSNN) with trainable kernel activation functions, where the parameters—kernel widths and center locations—are automatically determined via an alternating dual-staged iterative training procedure from our prior work [4].

In the *final* reduced tensor representation, the variable staying uncompressed is viewed as the *target variable*, whereas the variable compressed in the final stage of the MSTR procedure is referred to as the *secondary target variable*. To build a reduced-order model capable of providing predictions at new locations of both the parameter and time, it is necessary to ensure that they correspond to either the *target variable* or the *secondary target variable*. Upon ascertaining this, another interpolation/regression map $\mathcal{M}_{\text{basis}}$ is created, which can provide an approximation of the truncated orthogonal tensor appearing in the second-last stage of the MSTR procedure, i.e., $\tilde{\mathcal{L}}_{n_s-1}$, at new locations of the *secondary target variable*. The approximation from $\mathcal{M}_{\text{basis}}$ is denoted as $\hat{\tilde{\mathcal{L}}}_{n_s-1}$.

For this variant of our reduced-order model, the online phase involves querying $\mathcal{M}_{\text{tv}}$ to obtain $\hat{\mathcal{A}}_{n_s}$. This is then used, along with $\tilde{\mathcal{L}}_{n_s}$, to construct $\hat{\mathcal{A}}_{n_s-1}$. Later, yet another map $\mathcal{M}_{\text{stv}}$ is constructed

by interpolation/regression between specific slices of $\hat{\mathcal{A}}_{n_s-1}$, capable of providing an approximation of the *intermediate* reduced tensor representation from the second-last stage, corresponding to new locations of the *secondary target variable*. We denote this approximation by $\hat{\mathcal{A}}_{n_s-1}$. Next, $\hat{\mathcal{L}}_{n_s-1}$ is obtained by querying $\mathcal{M}_{\text{basis}}$, and in conjugation with $\hat{\mathcal{A}}_{n_s-1}$, an approximation of the *intermediate* reduced tensor representation from the third-last stage is constructed. By using this approximation in conjugation with the truncated orthogonal tensors $\{\tilde{\mathcal{L}}_i\}_{i=1}^{n_s-2}$, we obtain the approximation of the high-dimensional solution tensor at new locations of the *target variable* and *secondary target variable*, i.e., parameter and time. We create $\mathcal{M}_{\text{basis}}$ and $\mathcal{M}_{\text{stv}}$ using KSNNs, which is especially useful for efficiently constructing $\mathcal{M}_{\text{stv}}$ during the online phase. Moreover, an accurate construction of $\mathcal{M}_{\text{basis}}$ is non-trivial, requiring interpolation over a Grassmann manifold. We investigate several approaches to accomplish this.

To train the MSTR-based reduced-order model, we need the solution tensor $\mathcal{S} \in \mathbb{R}^{N_x \times N_y \times N_z \times m \times N_t}$, which requires data from either physical measurements or high-fidelity simulations across N_t time instances and m parameter configurations. This can be challenging if obtaining spatio-temporal solution fields for many parameters is infeasible or computationally expensive. To address this, we develop a method to progressively expand the solution tensor $\mathcal{S}$ along the parameter dimension from a small initial value m_0 to a moderate final value m_{final} . This incremental-learning procedure involves iterative applications of the MSTR strategy to create a surrogate $\hat{\mathcal{S}} \in \mathbb{R}^{N_x \times N_y \times N_z \times m_{\text{fine}} \times N_t}$, where the growth of $\mathcal{S}$ is guided by iteratively applying our t-EIM [5] over cheaply computable $\hat{\mathcal{S}}$ to extract critical parameter locations from a fine candidate set with cardinality m_{fine} . Moreover, employing randomized t-SVD for all factorizations in the MSTR procedure further enhances efficiency during the offline phase. Related to our t-EIM are the recently proposed tensor discrete empirical interpolation methods [1, 2] that use t-SVD to get the interpolation basis. In [1], a greedy procedure is used to pick the interpolation indices, while [2] uses the pivoted t-QR decomposition [3]. In contrast, t-EIM employs a greedy procedure to select both the interpolation indices and the basis.

We have observed excellent performance of the proposed framework over numerous high-dimensional tensor-valued datasets, comprising climate measurements as well as various parametric spatio-temporal flow phenomena exhibiting rich dynamics, including convection-dominated behavior. In the talk, we primarily intend to highlight the theoretical and algorithmic contributions of our work. Furthermore, we will illustrate the robustness of the MSTR strategy and the reduced-order model based on it over an appropriately selected numerical example.

Numerical results: Table 1 provides the configurations of selected order-3 and order-4 tensor datasets, detailing the training set dimensions, representing about half of the parameter samples and time steps; the rest form the test sets. Table 3 lists the average relative errors for tensor approximations using the MSTR procedure and the MSTR-based reduced-order model. To demonstrate the MSTR procedure’s advantage, Table 3 also includes average relative errors when the tensor is matricized to $S \in \mathbb{R}^{N_x N_y \times m N_t}$, thereby obtaining the truncated left singular matrix in $\mathbb{R}^{N_x N_y \times r}$ via

Tensor datasets	Spatial dimension(s)	# parameter samples	# time steps
Wave equation	$N = 10201$	$m = 19$	$N_t = 401$
Burgers’ equation	$N_x = 161, N_y = 161$	$m = 34$	$N_t = 101$
Navier-Stokes equations	$N = 37514$	$m = 18$	$N_t = 201$

Table 1: Details about the dimensions of selected order-3 and order-4 tensor datasets. All the examples have a 2D spatial domain with N denoting the total number of unstructured grid nodes.

Tensor datasets	SVD		MSTR		% drop
	Matricized	Compressed	Original	Compressed	
Wave equation	$\mathbb{R}^{10201 \times 7619}$	$\mathbb{R}^{10 \times 7619}$	$\mathbb{R}^{10201 \times 19 \times 401}$	$\mathbb{R}^{10 \times 19 \times 10}$	97.51%
Burgers' equation	$\mathbb{R}^{25921 \times 3434}$	$\mathbb{R}^{60 \times 3434}$	$\mathbb{R}^{161 \times 34 \times 161 \times 101}$	$\mathbb{R}^{10 \times 34 \times 10 \times 10}$	83.49%
Navier-Stokes equations	$\mathbb{R}^{37514 \times 3618}$	$\mathbb{R}^{10 \times 3618}$	$\mathbb{R}^{37514 \times 18 \times 201}$	$\mathbb{R}^{10 \times 18 \times 10}$	95.02%

Table 2: Details about the achieved level of compression for SVD and MSTR. The last column highlights % reduction in the entries of the *final* reduced tensor representation from MSTR in comparison with the compression achieved via SVD. The corresponding errors are reported in Table 3.

Tensor datasets	SVD	SVD-based ROM	MSTR	MSTR-based ROM
Wave equation	3.91×10^{-3}	4.08×10^{-3}	6.98×10^{-4}	1.21×10^{-3}
Burgers' equation	1.14×10^{-2}	8.61×10^{-2}	5.45×10^{-3}	6.14×10^{-3}
Navier-Stokes equations	2.25×10^{-2}	2.26×10^{-2}	4.29×10^{-4}	6.38×10^{-4}

Table 3: An illustrative comparison between average relative errors of the SVD, MSTR, and their respective reduced-order model (ROM) approximations over the test sets for selected datasets.

its SVD, projecting the matricized tensor on it, and producing the compressed representation in $\mathbb{R}^{r \times mN_t}$, where $r \ll mN_t$. The comparison between the results from SVD and MSTR is for equivalent compressions of the spatial dimensions. Table 2 details the compression levels, showing that the representation from MSTR possesses fewer total entries than the representation from SVD.

References

- [1] S. Ahmadi-Asl, A.-H. Phan, C. F. Caiafa, and A. Cichocki. Robust low tubal rank tensor recovery using discrete empirical interpolation method with optimized slice/feature selection. *Advances in Computational Mathematics*, 50(2):23, 2024. doi:10.1007/s10444-024-10117-8.
- [2] S. Chellappa, L. Feng, and P. Benner. Discrete empirical interpolation in the tensor t-product framework. e-prints 2410.14519v1, arXiv, 2024. doi:10.48550/arXiv.2410.14519.
- [3] N. Hao, M. E. Kilmer, K. Braman, and R. C. Hoover. Facial recognition using tensor-tensor decompositions. *SIAM Journal on Imaging Sciences*, 435(1):437–463, 2013. doi:10.1137/110842570.
- [4] H. Kapadia, L. Feng, and P. Benner. Active-learning-driven surrogate modeling for efficient simulation of parametric nonlinear systems. *Computer Methods in Applied Mechanics and Engineering*, 419:116657, 2024. doi:10.1016/j.cma.2023.116657.
- [5] H. Kapadia, L. Feng, and P. Benner. Data-driven optimal sensor placement for tensorial data reconstruction via the tensor t-product framework. Technical report, 2024. In preparation.
- [6] M. E. Kilmer, K. Braman, N. Hao, and R. C. Hoover. Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging. *SIAM Journal on Matrix Analysis and Applications*, 34(1):148–172, 2013. doi:10.1137/110837711.
- [7] M. E. Kilmer and C. D. Martin. Factorization strategies for third-order tensors. *Linear Algebra and its Application*, 435(3):641–658, 2011. Special issue: Dedication to Pete Stewart on the occasion of his 70th birthday. doi:10.1016/j.laa.2010.09.020.
- [8] C. D. Martin, R. Shafer, and B. LaRue. An order- p tensor factorization with applications in imaging. *SIAM Journal on Scientific Computing*, 35(1):A474–A490, 2013. doi:10.1137/110841229.
- [9] J. Zhang, A. K. Saibaba, M. E. Kilmer, and S. Aeron. A randomized tensor singular value decomposition based on the t-product. *Numerical Linear Algebra with Applications*, 25(5):e2179, 2018. doi:10.1002/nla.2179.