Implicit Regularization in Deep Tucker Factorization: Low-Rankness via Structured Sparsity

Kais Hariz[1,2] **Hachem Kadri**[1] **Stêphane Ayache**[1] **Maher Moakher**[2] **Thierry Artières**[1,3]
[1]Aix Marseille University, CNRS, LIS, Marseille, France
[2]LAMSIN, National Engineering School of Tunis, University of Tunis El Manar, Tunis, Tunisia
[3]Ecole Centrale de Marseille, Marseille, France

Abstract

We theoretically analyze the implicit regularization of deep learning for tensor completion. We show that deep Tucker factorization trained by gradient descent induces a structured sparse regularization. This leads to a characterization of the effect of the depth of the neural network on the implicit regularization and provides a potential explanation for the bias of gradient descent towards solutions with low multilinear rank. Numerical experiments confirm our theoretical findings and give insights into the behavior of gradient descent in deep tensor factorization.

1 INTRODUCTION

The question how deep neural networks generalize well has not been answered; it continues to stimulate research and generate hypotheses (Zhang et al., 2021; Belkin et al., 2019; Bartlett et al., 2020). An interesting line of research suggests that implicit regularization could provide a means to understand the underlying mechanisms behind the ability of neural networks to generalize even when the number of learning parameters is much larger than the number of training examples (Neyshabur et al., 2014). This paper pursues this line of attack by characterizing the implicit regularization in deep Tucker tensor factorization.

A number of studies have investigated the role of implicit regularization in deep learning (see Gunasekar et al., 2017; Arora et al., 2019; Chizat and Bach, 2020; Razin and Cohen, 2020; Li et al., 2021a; Gissin et al., 2020; Li et al., 2021b; Milanesi et al., 2021; Razin et al., 2021; Ge et al., 2021; Ziyin, 2023 and references therein). Among these works, two papers have particularly attracted our attention. First, Arora et al. (2019) considered deep matrix factorization in overparameterized regime and showed that even when the rank of the factorized matrix is not constrained, adding depth to matrix factorization promotes the convergence of gradient descent to low-rank solutions. This was observed empirically and also analyzed theoretically through the characterization of the singular value dynamics during training. Then, Razin et al. (2021) pushed the analysis further and extended it to tensor factorization. They considered Canonical-Polyadic (CP) factorization which are based on decomposing a tensor into the sum of rank-one tensors. The canonical rank of a tensor is the minimum number of rank-one tensors of the CP decomposition. Razin et al. (2021) studied the dynamics of the norms of these rank-one tensors and showed that under some conditions, gradient descent over CP factorization exhibits an implicit regularization towards low tensor rank. This means that only a small number of rank-one tensors will emerge (i.e., will not be too close to zero) after learning.

Expanding the analysis of implicit regularization from matrix factorization to tensor factorization is important as it has the potential to go beyond linear neural networks. Indeed, previous work showed connections between tensor decompositions and certain types of nonlinear neural networks (Cohen et al., 2016; Razin et al., 2021). Recently, Hariz et al. (2022) considered a generalized overparameterized CP decomposition, which can be viewed as a deep extension of the standard CP tensor model, and showed that the effect of the implicit regularization towards low CP rank in deep tensor CP factorization via gradient descent may grow polynomially with the depth of the factorization. In the same line, Razin et al. (2022) considered hierarchical Tucker (HT) factorization. This choice was motivated by the fact that HT factorization corresponds to a certain deep convolutional neural network (Cohen et al., 2016). Similar to the CP decompostion, they

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238. Copyright 2024 by the author(s).

established implicit regularization of gradient descent towards low hierarchical Tucker rank. To complete the picture, we focus in this work on Tucker factorization optimized through gradient-based methods and the induced implicit regularization.

In this paper, we make the following contributions:

- we analyze overparameterized Tucker tensor factorization and theoretically characterize its evolution during gradient-descent learning,
- we prove that deep Tucker factorization trained by gradient descent induces a structured sparse regularization,
- we show how this sparse implicit regularization promotes low-rank solutions,
- we conduct numerical experiments that confirm our theoretical findings, offering insights into the behavior of gradient descent in deep Tucker factorization.

It is worth noting that the convergence of gradient-descent in overparameterized Tucker factorization to solutions with low tensor rank has been empirically observed in Milanesi et al. (2021) but was not theoretically investigated. Previous work which has theoretically studied implicit regularization in tensor factorization focused on the canonical rank and the hierarchical rank (Razin et al., 2021; Ge et al., 2021; Hariz et al., 2022; Razin et al., 2022). In this work, we consider overparameterized Tucker factorization and so interested in the multilinear rank of a tensor. Finally, a recent paper Li et al. (2023) asked the question whether gradient descent can induce other forms of implicit regularization than ℓ_2-norm, sparsity and low-rankness, and showed for a certain type of networks, called diagonally grouped linear neural network, that it biases towards solutions with a group sparsity structure. From this point of view our results show a similar behavior in overparameterized Tucker factorization. To the best of our knowledge, this has not been previously reported.

2 PRELIMINARIES

In this section we introduce tensors and the related multi-linear algebra. For a more comprehensive account on this subject we refer the reader to Kolda and Bader (2009). We use the notation $[\![1, N]\!]$ for the set $\{1, \dots, N\}$. We will denote vectors by lowercase letters, e.g. $v \in \mathbb{R}^{I_1}$ whereas matrices and tensors will be denoted respectively by uppercase and calligraphic letters, e.g. $M \in \mathbb{R}^{I_1 \times I_2}$ and $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. We use $\|\cdot\|$ to denote the Euclidean norm of the vectors. For a matrix M, $\|M\|$ denotes its Frobenius norm and $\|M\|_2$ its spectral norm, i.e., the largest singular value. The Kronecker product of two matrices $M \in \mathbb{R}^{I_1 \times I_2}$ and $N \in \mathbb{R}^{J_1 \times J_2}$ is denoted by $M_1 \odot M_2 \in \mathbb{R}^{I_1 J_1 \times I_2 J_2}$. The $(i_1, i_2, \dots, i_N)$-th entry of $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ will be denoted by $\mathcal{T}_{i_1, i_2, \dots, i_N}$ where $i_n = 1, 2, \dots, I_n$, for all $n \in [\![1, N]\!]$. Given two tensors $\mathcal{T}, \mathcal{S} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ their scalar product writes

$$\langle \mathcal{T}, \mathcal{S} \rangle = \sum_{i_1} \sum_{i_2} \cdots \sum_{i_N} \mathcal{S}_{i_1, i_2 \cdots i_N} \mathcal{T}_{i_1, i_2 \cdots i_N},$$

and the Frobenius norm of $\mathcal{T}$ is defined as $\|\mathcal{T}\| = \sqrt{\langle \mathcal{T}, \mathcal{T} \rangle}$.

We denote by $\mathcal{T}_{:,\dots,:,i_m,:,\dots,:}$ the subtensor of $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ obtained by fixing the m-th index of the tensor $\mathcal{T}$ with value equal to i_m and varying all the other indexes. For a 3-rd order tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, the subtensors are the slices of the tensor, which are the matrices obtained by fixing one index: as an example, $\mathcal{T}_{:i_2:} \in \mathbb{R}^{I_1 \times I_3}$ is the slice obtained by fixing the second index to i_2.

Given N vectors $v_1 \in \mathbb{R}^{I_1}, v_2 \in \mathbb{R}^{I_2}, \dots, v_N \in \mathbb{R}^{I_N}$, their outer product is the tensor whose $(i_1, \dots, i_N)$-th entry writes $(v_1 \otimes v_2 \otimes \cdots \otimes v_N)_{i_1, i_2 \dots, i_N} = (v_1)_{i_1} (v_2)_{i_2} \dots (v_N)_{i_N}$ for all $i_n \in [1, \dots, I_n]$, $n \in [1, \dots, N]$. An N-th order tensor $\mathcal{T}$ is called a rank-1 tensor if it can be written as the outer product of N vectors, i.e. $\mathcal{T} = v_1 \otimes v_2 \otimes \cdots \otimes v_N$. In this work, we focus on the multilinear rank of a tensor. It is based on unfolding a tensor along each of its modes to obtain matricizations. The n-unfolding of a tensor is defined as follows: Given an N-th order tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, the matrix $T_{(n)} \in \mathbb{R}^{I_n \times (I_1 I_2 \dots I_{n-1} I_{n+1} I_{n+2} \dots I_N)}$ is called the unfolding of $\mathcal{T}$ along the n-th mode. This bring us to define the n-rank and the multilinear rank.

Definition 2.1. The tensor n-rank of $\mathcal{T}$ is defined as the rank of $T_{(n)}$. The multilinear rank, also called Tucker rank, is defined as the N-tuple whose i-th entry is the i-th rank of $\mathcal{T}$.

The multilinear rank is intimately related to the higher-order singular value decomposition. The first step to take is to define a proper way to multiply a tensor and a matrix.

Definition 2.2. The n-mode product of a tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ with a matrix $U \in \mathbb{R}^{J \times I_n}$ is a tensor of size $(I_1 \times I_2 \times \cdots I_{n-1} \times J \times I_{n+1} \cdots \times I_N)$ denoted by $\mathcal{T} \times_n U$ and its entries are defined as

$$(\mathcal{T} \times_n U)_{i_1 \cdots i_{n-1} j i_{n+1} \cdots i_N} = \sum_{i_n = 1}^{I_n} \mathcal{T}_{i_1 \cdots i_{n-1} i_n i_{n+1} \cdots i_N} U_{j i_n}.$$

The n-mode product satisfies to the following property: given $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and two matrices $F \in \mathbb{R}^{J \times I_n}$,

$G \in \mathbb{R}^{K\times J}$, one has

$$(\mathcal{T} \times_n F) \times_n G = \mathcal{T} \times_n (G \cdot F)\,,$$

which generalizes to more than two matrices. We are in shape to write a Singular Value Decomposition (SVD) applying to tensors, named Higher-Order Singular Value Decomposition (HOSVD).

Theorem 2.3 (HOSVD De Lathauwer et al., 2000)**.** *Any tensor $\mathcal{T} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ can be written as the product*

$$\mathcal{T} = \mathcal{C} \times_1 U^{(1)} \times_2 U^{(2)} \cdots \times_N U^{(N)},$$

where

1. *$U^{(n)}$ is an orthogonal $(I_n \times I_n)$-matrix, usually denoted as factor matrix;*
2. *$\mathcal{C}$ is a $(I_1 \times I_2 \times \cdots I_N)$-tensor, usually referred to as core tensor, of which the subtensors $\mathcal{C}_{i_n=\alpha}$ obtained by fixing the n-th index to α are such that*
   - *(i) they are mutually orthogonal: given two subtensors $\mathcal{C}_{i_n=\alpha}$ and $\mathcal{C}_{i_n=\beta}$, they are orthogonal for all possible values of n, α, β subject to $\alpha \neq \beta$:*
   $$\langle \mathcal{C}_{i_n=\alpha}, \mathcal{C}_{i_n=\beta}\rangle = 0,$$
   - *(ii) they are ordered:*
   $$\|\mathcal{C}_{i_n=1}\| \geq \|\mathcal{C}_{i_n=2}\| \cdots \geq \|\mathcal{C}_{i_n=I_n}\| \geq 0.$$

The analogy with matrix SVD is straightforward: the Frobenius norm of subtensors $\|\mathcal{C}_{i_n=i}\| := \sigma_i^{(n)}$ are the n-mode singular values and the column vectors of the matrices $U^{(n)}$ are the n-mode singular vectors. As in the matrix case, where the number of non-zero singular values controls the rank, in the higher-order setup we have that, if R_n is equal to the highest index for which $\|\mathcal{C}_{i_n=R_n}\| > 0$ then one has that the n-rank of $\mathcal{T}$ is equal to R_n. Tucker decomposition (Tucker, 1966) with rank $(R_1, R_2, \ldots, R_N)$ can be obtained from HOSVD by considering only the first R_n column vectors of the matrix $U^{(n)}$ for each $n \in [1, \ldots, N]$.

3 DEEP TUCKER FACTORIZATION

Following the terminology in Jiang et al. (2017), a Tucker decomposition of a tensor $\mathcal{W} \in \mathbb{R}^{d_1\times\cdots\times d_N}$ is the factorization of $\mathcal{W}$ as the product of a core tensor $\mathcal{G}$ and N factor matrices $V^{(n)}$, $n = 1, \ldots, N$, i.e.,

$$\mathcal{W} = \mathcal{G} \times_1 V^{(1)} \times_2 V^{(2)} \cdots \times_N V^{(N)}. \tag{1}$$

A Tucker decomposition is said to be orthonormal if each factor matrix has orthonormal columns. The orthonormal Tucker decomposition corresponds to the HOSVD (see Theorem 2.3). In the literature, the term Tucker decomposition generally refers to the orthonormal Tucker decomposition.

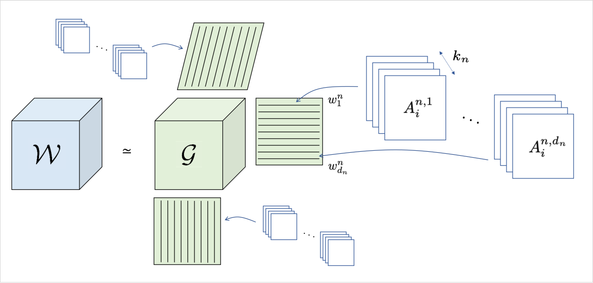

Figure 1: Overparameterized Deep Tucker Factorization

Tensor completion and overparameterized Tucker factorization The learning problem we consider is tensor completion. Let $\mathcal{A} \in \mathbb{R}^{d_1\times\cdots\times d_N}$ be the tensor with missing entries to be recovered. We denote by $\Omega \subset [\![1, d_1]\!] \times \ldots \times [\![1, d_N]\!]$ the set of indexes with non-zero (i.e., observed) entries. We consider a tensor learning problem based on minimizing the following objective function:

$$\mathcal{L}(\mathcal{W}) := \frac{1}{|\Omega|} \sum_{(i_1,\ldots,i_N)\in\Omega} \ell(\mathcal{W}_{i_1,\ldots,i_N} - \mathcal{A}_{i_1,\ldots,i_N}), \tag{2}$$

where ℓ is a differentiable and locally smooth loss function.

Our goal is to study implicit regularization in overparameterized regimes. So, the tensor $\mathcal{W}$ is overparameterized using a deep Tucker decomposition. More formally, $\mathcal{W}$ has the same form as in (1) with a core tensor $\mathcal{G} \in \mathbb{R}^{d_1\times\cdots\times d_N}$, which has the same dimension as to the original tensor $\mathcal{W}$, and factor matrices $V^{(n)} \in \mathbb{R}^{d_n\times d_n}$, $\forall n \in [\![1, N]\!]$, which are also overparameterized via a product of multiple matrices, i.e., $V^{(n)} = \left[\prod_{i=1}^{k_n} A_i^{n,1}\omega_1^n, \ldots, \prod_{i=1}^{k_n} A_i^{n,d_n}\omega_{d_n}^n\right]$, where $A_i^{n,r_n} \in \mathbb{R}^{d_n\times d_n}$ and $\omega_{r_n}^n \in \mathbb{R}^{d_n}$, $\forall i \in [\![1, k_n]\!]$ and $\forall r_n \in [\![1, d_n]\!]$. The core tensor $\mathcal{G}$, the matrices A_i^{n,r_n} and the weight vectors $w_{r_n}^n$ are the parameters to be learned. $k_n + 1$ is the depth along the n-th mode. Figure 1 schematically shows the overparameterized Tucker factorization. Note that the deep tucker decomposition of $\mathcal{W}$ can also be written as follows

$$\mathcal{W} = \sum_{r_1=1}^{d_1} \cdots \sum_{r_N=1}^{d_N} \mathcal{G}_{r_1,\ldots,r_N} \bigotimes_{n=1}^{N} \prod_{i=1}^{k_n} A_i^{n,r_n}\omega_{r_n}^n.$$

If $\mathcal{G}_{r_1,\ldots,r_N} = 0$ unless $r_1 = \ldots = r_N$, we obtain the deep CP decomposition (Hariz et al., 2022).

In the following, we consider learning the overparameterized Tucker factorization of $\mathcal{W}$ by minimizing (2) using gradient descent. Let $\mathcal{L}(\mathcal{W}) =$

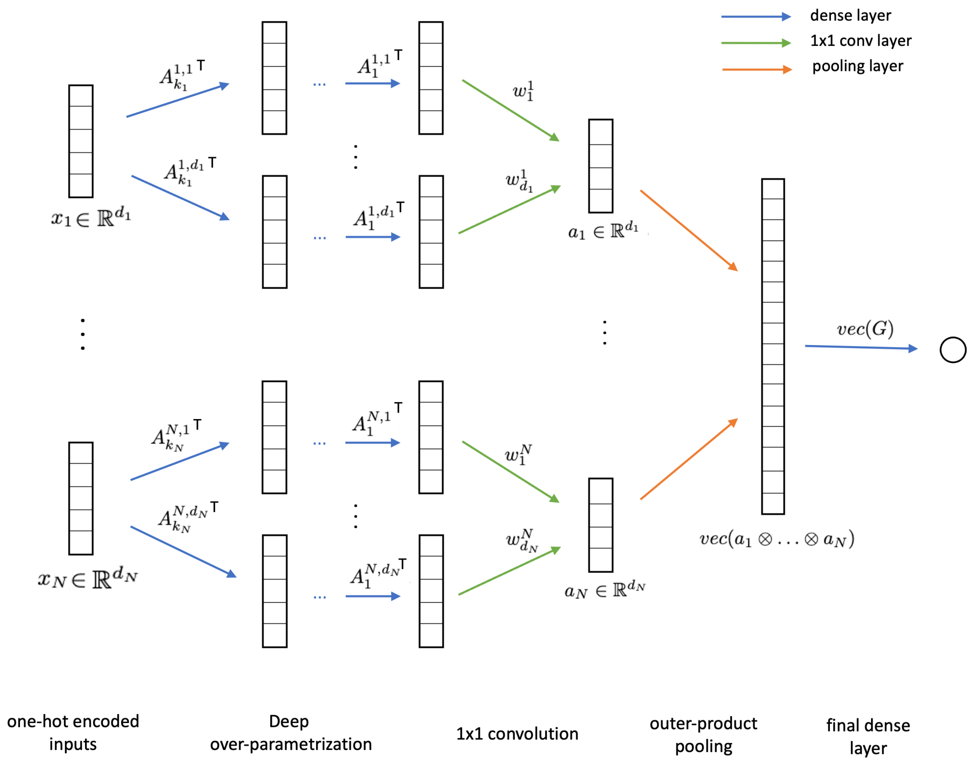

Figure 2: Neural network corresponding to deep Tucker model. It is composed of: representation, 1×1 convolution, outer-product pooling and dense layers.

$\Phi\Big(\{\omega_{r_n}^n\}_{n,r_n}, \{A_i^{n,r_n}\}_{n,r_n,i}, \{\mathcal{G}_{r_1,\dots,r_N}\}_{r_1,\dots,r_N}\Big)$, $\forall n \in [\![1, N]\!], \forall r_n \in [\![1, d_n]\!], \forall i \in [\![1, k_n]\!]$. Following previous work on implicit regularization in matrix and tensor factorization (Arora et al., 2019; Razin et al., 2021), we consider gradient flow, which can be viewed as the limit of gradient descent for infinitesimally small learning rates:

$$\frac{d}{dt}\omega_{r_m}^m(t) = -\frac{\partial}{\partial \omega_{r_m}^m}\Phi\Big(\{\omega_{r_n}^n(t)\}, \{A_i^{n,r_n}(t)\}, \{\mathcal{G}_{r_1,\dots,r_N}(t)\}\Big),$$

$$\frac{d}{dt}A_j^{m,r_m}(t) = -\frac{\partial}{\partial A_j^{m,r_m}}\Phi\Big(\{\omega_{r_n}^n(t)\}, \{A_i^{n,r_n}(t)\}, \{\mathcal{G}_{r_1,\dots,r_N}(t)\}\Big),$$

and

$$\frac{d}{dt}\mathcal{G}_{r_1,\dots,r_N}(t) = -\frac{\partial}{\partial \mathcal{G}_{r_1,\dots,r_N}}\Phi\Big(\{\omega_{r_n}^n(t)\}, \{A_i^{n,r_n}(t)\}, \{\mathcal{G}_{s_1,\dots,s_N}(t)\}\Big).$$

Our aim is to characterize the dynamics during gradient flow of the norms of the subtensors $\{\mathcal{G}_{:\dots:r_n:\dots:}\}_{r_n}$ of the core tensor $\mathcal{G}$, as well as those of the norms of the weight matrices $\{A_i^{n,r_n}\}_{n,r_n,i}$ and vectors $\{\omega_{r_n}^n\}_{n,r_n}$.

Connection to deep neural networks Cohen et al. (2016) showed that tensor completion using CP tensor decomposition is equivalent to solving a prediction task with a certain type of neural networks, consisting of a representation layer which is followed by a single hidden layer and the output layer (see also Razin et al., 2021). From this point of view, the overparameterized Tucker factorization can also be represented by a neural network. As in deep CP (Hariz et al., 2022), the neural network equivalent to tensor completion using deep Tucker factorization adds depth to the representation layer using the matrices A_i^{n,r_n}. More formally, the deep Tucker neural network computes the function:

$$f(x_1,\dots,x_N) := \sum_{r_1,\dots,r_N} G_{r_1,\dots,r_N}\prod_{n=1}^{N}\left\langle w_{r_n}^n, \left(\prod_{j=1}^{k_n} A_j^{n,r_n}\right)^\top x_n\right\rangle.$$

It consists of:

- a representation layer: a (deep) series of dense layers using weight matrices A_j^{n,r_n} that computes a new representation of x_n, i.e., $f_{r_n}^n(x_n) := \prod_{1}^{j=k_n}\left(A_j^{n,r_n}\right)^\top x_n$, $\forall n \in [\![1, N]\!], \forall r_n \in [\![1, d_n]\!], \forall x_n \in \mathbb{R}^{d_n}$;
- a hidden layer which is composed of:

1. a 1×1 convolution operator with learnable filters $\{w^n_{r_n}\}_{n=1}^N$ which computes the vectors $a_n \in \mathbb{R}^{d_n}$ such that $(a_n)_{r_n} = \langle w^n_{r_n}, f^n_{r_n}(x_n)\rangle$, $\forall n \in [\![1, N]\!], r_n \in [\![1, d_n]\!]$;
2. a global pooling that outputs pool(:) $=$ vec$(a_1 \otimes \ldots \otimes a_N)$—this corresponds to a product pooling operation which computes $\prod_{n=1}^N \langle w^n_{r_n}, f^n_{r_n}(x_n)\rangle$, $(r_1, \ldots, r_N) \in [\![1, d_1]\!] \times \ldots \times [\![1, d_N]\!]$;

- a fully connected layer with weights $G_{r_1,\ldots,r_N}$ to compute the output $\langle \text{vec}(G), \text{vec}(a_1 \otimes \ldots \otimes a_N)\rangle = \sum_{r_1,\ldots,r_N} G_{r_1,\ldots,r_N} \prod_{n=1}^N \langle w^n_{r_n}, f^n_{r_n}(x_n)\rangle$.

In the case of tensor completion, the input of this neural network is the multi-index $(i_1, \ldots, i_N) \in [d_1] \times \ldots [d_N]$ encoded via one hot encoding $(x_1, \ldots, x_N) \in \mathbb{R}^{d_1} \times \ldots \mathbb{R}^{d_N}$. The output is

$$W_{i_1,\ldots,i_N} = \sum_{r_1,\ldots,r_N} G_{r_1,\ldots,r_N} \prod_{n=1}^N \left(\prod_{j=1}^{k_n} A_j^{n,r_n} w^n_{r_n} \right)_{i_n}$$

$$= \sum_{r_1,\ldots,r_N} G_{r_1,\ldots,r_N} \prod_{n=1}^N \left\langle \prod_{j=1}^{k_n} A_j^{n,r_n} w^n_{r_n}, x_n \right\rangle = f(x_1, \ldots, x_N).$$

Figure 2 provides a schematic representation of the neural network corresponding to the deep Tucker factorization model.

4 THEORETICAL ANALYSIS

This section contains the main theoretical results of the paper. All the proofs are given in the supplementary material.

4.1 Dynamics of Gradient Flow Over Deep Tucker Factorization

Before stating our main theorem, we need the following lemma.

Lemma 4.1. *For all $m \in [\![1, N]\!]$, $r_m \in [\![1, d_m]\!]$, $i, j \in [\![1, k_m]\!]$, the following hold $\forall t \geq 0$*

i. $\|\omega^m_{r_m}(t)\|^2 - \|\mathcal{G}_{:\ldots:r_m:\ldots:}(t)\|^2 = \|\omega^m_{r_m}(0)\|^2 - \|G_{:\ldots:r_m:\ldots:}(0)\|^2$,

ii. $\|A_i^{m,r_m}(t)\|^2 - \|\omega^m_{r_m}(t)\|^2 = \|A_i^{m,r_m}(0)\|^2 - \|\omega^m_{r_m}(0)\|^2$,

iii. $\|A_i^{m,r_m}(t)\|^2 - \|A_j^{m,r_m}(t)\|^2 = \|A_i^{m,r_m}(0)\|^2 - \|A_j^{m,r_m}(0)\|^2$.

We proved this result by showing that, $\forall m \in [\![1, N]\!], \forall r_m \in [\![1, d_m]\!], \forall i \in [\![1, k_m]\!], \forall t \geq 0$,

$$\frac{d}{dt}\|\omega^m_{r_m}(t)\|^2 = \frac{d}{dt}\|\mathcal{G}_{:\ldots:r_m:\ldots:}(t)\|^2 = \frac{d}{dt}\|A_i^{m,r_m}(t)\|^2.$$

Lemma 4.1 shows that for every column vector $w^m_{r_m}$, the corresponding subtensor $\mathcal{G}_{:\ldots:r_m:\ldots:}$ of the core tensor $\mathcal{G}$ satisfies that $\|\mathcal{G}_{:\ldots:r_m:\ldots:}(t)\|^2 - \|\omega^m_{r_m}(t)\|^2$ is constant over time. So, the Frobenius norms of the subtensors of the core tensor in the m-th mode can be obtained from the ℓ_2-norms of the column vectors. If at initialization $\|\mathcal{G}_{:\ldots:r_m:\ldots:}(0)\|^2 = \|\omega^m_{r_m}(0)\|^2$, then $\|\mathcal{G}_{:\ldots:r_m:\ldots:}(t)\|^2 = \|\omega^m_{r_m}(t)\|^2$, $\forall t \geq 0$. This time-invariant dynamic pattern also holds for the weight matrices A_i^{m,r_m}.

Lemma 4.1 leads to a notion of unbalancedness magnitude for the deep Tucker factorization, as in previous work on matrix and tensor decomposition (Arora et al., 2019; Razin et al., 2021; Hariz et al., 2022; Razin et al., 2022).

Definition 4.2. The unbalancedness magnitude at time $t \geq 0$ of deep Tucker factorization is defined as:

$$\varepsilon(t) = \max\left(\varepsilon_1(t), \varepsilon_2(t)\right),$$

where

$$\varepsilon_1(t) = \max_{\substack{m=1\ldots N \\ r_m=1\ldots d_m}} \left| \|\omega^m_{r_m}(t)\|^2 - \|\mathcal{G}_{:\ldots:r_m:\ldots:}(t)\|^2 \right|,$$

and

$$\varepsilon_2(t) = \max_{\substack{m=1\ldots N \\ r_m=1\ldots d_m \\ i=1\ldots k_m}} \left| \|\omega^m_{r_m}(t)\|^2 - \|A_i^{m,r_m}(t)\|^2 \right|.$$

By Lemma 4.1, the unbalancedness magnitude stays constant during gradient descent training. So, if at initialization $\varepsilon(0) = 0$, the unbalancesdness magnitude at time t, $\varepsilon(t)$ remains zero. This leads to the fact that, $\forall m \in [\![1, N]\!], \forall r_m \in [\![1, d_m]\!], \forall i \in [\![1, k_m]\!]$,

$$\|\omega^m_{r_m}(t)\| = \|\mathcal{G}_{:\ldots:r_m:\ldots:}(t)\| = \|A_i^{m,r_m}(t)\|. \quad (3)$$

We are now able to state our main result.

Theorem 4.3. *Assume that $\varepsilon(0) = 0$. Then, for any $m \in [\![1, N]\!]$, $r_m \in [\![1, d_m]\!]$ and time $t \geq 0$ at which $\prod_{n=1}^N \prod_{r_n=1}^{d_n} \left\| \prod_{j=1}^{k_n} A_j^{n,r_n}(t)\omega^n_{r_n}(t) \right\| \neq 0$, the subtensors of the core tensor and the weight vectors of the deep Tucker factorization evolve according to*

$$\frac{d}{dt}\|\mathcal{G}_{:\ldots:r_m:\ldots:}(t)\| = \frac{d}{dt}\|\omega^m_{r_m}(t)\| = \frac{\left\| \prod_{j=1}^{k_m} A_j^{m,r_m}(t)\omega^m_{r_m}(t) \right\|}{\|\omega^m_{r_m}(t)\|} \delta_{r_m}(t),$$

where

$$\delta_{r_m}(t) := \sum_{r_1,\ldots,r_{m-1},r_{m+1},\ldots,r_N} \mathcal{G}_{r_1,\ldots,r_N}(t) \left(\prod_{n \neq m} \left\| \prod_{j=1}^{k_n} A_j^{n,r_n}(t)\omega^n_{r_n}(t) \right\| \right) \Delta_{r_1,\ldots,r_N}(t)$$

with

$$\Delta_{r_1,\ldots,r_N}(t) := \left\langle -\nabla\mathcal{L}(\mathcal{W}(t)), \bigotimes_{n=1}^{N} \frac{\prod_{j=1}^{k_n} A_j^{n,r_n}(t)\omega_{r_n}^n(t)}{\left\|\prod_{j=1}^{k_n} A_j^{n,r_n}(t)\omega_{r_n}^n(t)\right\|}\right\rangle.$$

Theorem 4.3 provides a differential equation that characterizes the dynamics of gradient flow when solving a tensor completion problem with a deep Tucker factorization. A natural question is whether an implicit regularization arises from performing gradient descent and how it changes with the depth of the factorization, which we address next.

4.2 Implicit Bias Towards Structured Sparsity

We now study the effect of the depth on the evolution of the subtensor norms during gradient flow.

Corollary 4.4. *Under the same assumptions as in Theorem 4.3 and if the matrices* $\{A_i^{n,r_n}(0)\}_{n=1}^{N}{}_{r_n=1}^{d_n}{}_{i=1}^{k_n}$ *satisfy* $A_i^{n,r_n}(0)^\top A_i^{n,r_n}(0) = A_{i+1}^{n,r_n}(0)A_{i+1}^{n,r_n}(0)^\top$ *for all* $i \in [\![1, k_n-1]\!]$ *with* $k_n \geq 2$, $r_n \in [\![1, d_n]\!]$ *and* $n \in [\![1, N]\!]$, *we have:*

i. if $\frac{d}{dt}\|\mathcal{G}_{:\ldots:r_m:\ldots:}(t)\| \geq 0$, *then*

$$\left(\frac{\|\mathcal{G}_{:\ldots:r_m:\ldots:}(t)\|}{\sqrt{d_m}}\right)^{k_m} \alpha_{r_m}(t)\delta_{r_m}(t) \leq \frac{d}{dt}\|\mathcal{G}_{:\ldots:r_m:\ldots:}(t)\| \leq \|\mathcal{G}_{:\ldots:r_m:\ldots:}(t)\|^{k_m}\delta_{r_m}(t),$$

where $\alpha_{r_m}(t) := \left|\langle v_{r_m}^{1,m}(t), \hat{\omega}_{r_m}^m(t)\rangle\right|$ *with* $\hat{\omega}_{r_m}^m(t) := \frac{\omega_{r_m}^m(t)}{\|\omega_{r_m}^m(t)\|}$ *and* $v_{r_m}^{1,m}(t)$ *being the first left singular vector of* $A^{m,r_m}(t) := \prod_{i=1}^{k_m} A_i^{m,r_m}(t)$,

ii. if $\frac{d}{dt}\|\mathcal{G}_{:\ldots:r_m:\ldots:}(t)\| \leq 0$, *the inequalities are reversed,*

iii. if in addition $\prod_{n\neq m}\left\|\prod_{j=1}^{k_n} A_j^{n,r_n}(t)\omega_{r_n}^n(t)\right\| \leq 1$, *for all* $r_n \in [\![1, d_n]\!]$ *and* $n \neq m$, *then*

$$|\delta_{r_m}(t)| \leq M\|\mathcal{G}_{:\ldots:r_m:\ldots:}(t)\|,$$

with $M = \sup_{t\in[0,T]}\sqrt{\sum_{r_1,\ldots,r_{m-1},r_{m+1},\ldots,r_N} \Delta^2_{r_1,\ldots,r_N}(t)}$ *and training time* T.

The inequalities in the above corollary also hold when $k_n = 1$. Corollary 4.4 shows that the derivative of the subtensor norms is bounded by quantities which are proportional to their size raised to the power of the depth. As a result, the norms of the subtensors can move faster when large and slower when small, especially in situations where δ_{r_m} is dominated by the subtensor norms. This is ensured when the condition in (iii) of Corollary 4.4 is satisfied. This effect enhances the convergence to solutions with core tensors having a few number of subtensors with non-nonzero norms. A *structured sparsity* in the core tensor $\mathcal{G}$ is then promoted. To see this more clearly, consider a 3-rd order core tensor $\mathcal{G} \in \mathbb{R}^{d_1\times d_2\times d_3}$, the subtensors of $\mathcal{G}$ in the mode 2 are the slices (i.e., matrices) $\{\mathcal{G}_{:r_2:}\}_{r_2=1}^{d_2}$. The evolution rates of the norm of these slices are proportional to their norm raised to the power of k_2. When the depth k_2 increases, the norms of some slices, those which have the highest norms, will increase, while slices with very small norms will remain with small norm during training. At the end of the process, many entries of the core tensor $\mathcal{G}$ will be near to zero, and moreover these entries will be grouped into slices. This leads to a structured sparse core tensor $\mathcal{G}$. The same holds for the other modes of the tensor and this effect is more pronounced with larger depths. The same reasoning applies to vector norms $\|\omega_{r_n}^n(t)\|$ and matrix norms $\|A_i^{n,r_n}(t)\|$ since they evolve similarly over time (see (3)).

We now consider Tucker factorization without depth (i.e., $k_1 = \ldots = k_N = 0$).

Corollary 4.5. *Under the same assumptions as in Theorem 4.3 and if* $k_1 = \ldots = k_N = 0$, *we have*

$$\frac{d}{dt}\|\omega_{r_m}^m(t)\| = \sum_{r_1,\ldots,r_{m-1},r_{m+1},\ldots,r_N} \mathcal{G}_{r_1,\ldots,r_N}(t)\prod_{n\neq m}\left\|\omega_{r_n}^n(t)\right\|\hat{\Delta}_{r_1,\ldots,r_N}(t),$$

and

$$\frac{d}{dt}\|\mathcal{G}_{:\ldots:r_m:\ldots:}(t)\| = \sum_{r_1,\ldots,r_{m-1},r_{m+1},\ldots,r_N}\mathcal{G}_{r_1,\ldots,r_N}(t)\Big(\prod_{n\neq m}\|\mathcal{G}_{:\ldots:r_n:\ldots:}(t)\| \hat{\Delta}_{r_1,\ldots,r_N}(t)\Big),$$

where $\hat{\Delta}_{r_1,\ldots,r_N}(t) = \left\langle -\nabla\mathcal{L}(\mathcal{W}(t)), \bigotimes_{n=1}^{N}\frac{\omega_{r_n}^n(t)}{\|\omega_{r_n}^n(t)\|}\right\rangle.$

Corollary 4.5 shows that in contrast to the deep factorization model, the derivative of the norm of the subtensor $G_{:\ldots:r_m:\ldots:}$ in the case of Tucker decompostion without depth does not depend on its norm but only on the norms of the subtensors in the other modes. The norms of all subtensors interact with each other, and then the implicit bias towards sparse solutions is less visible.

4.3 Relation to Multilinear Rank

Since our task is tensor completion, one can ask whether the previous results on the dynamics of gradient flow

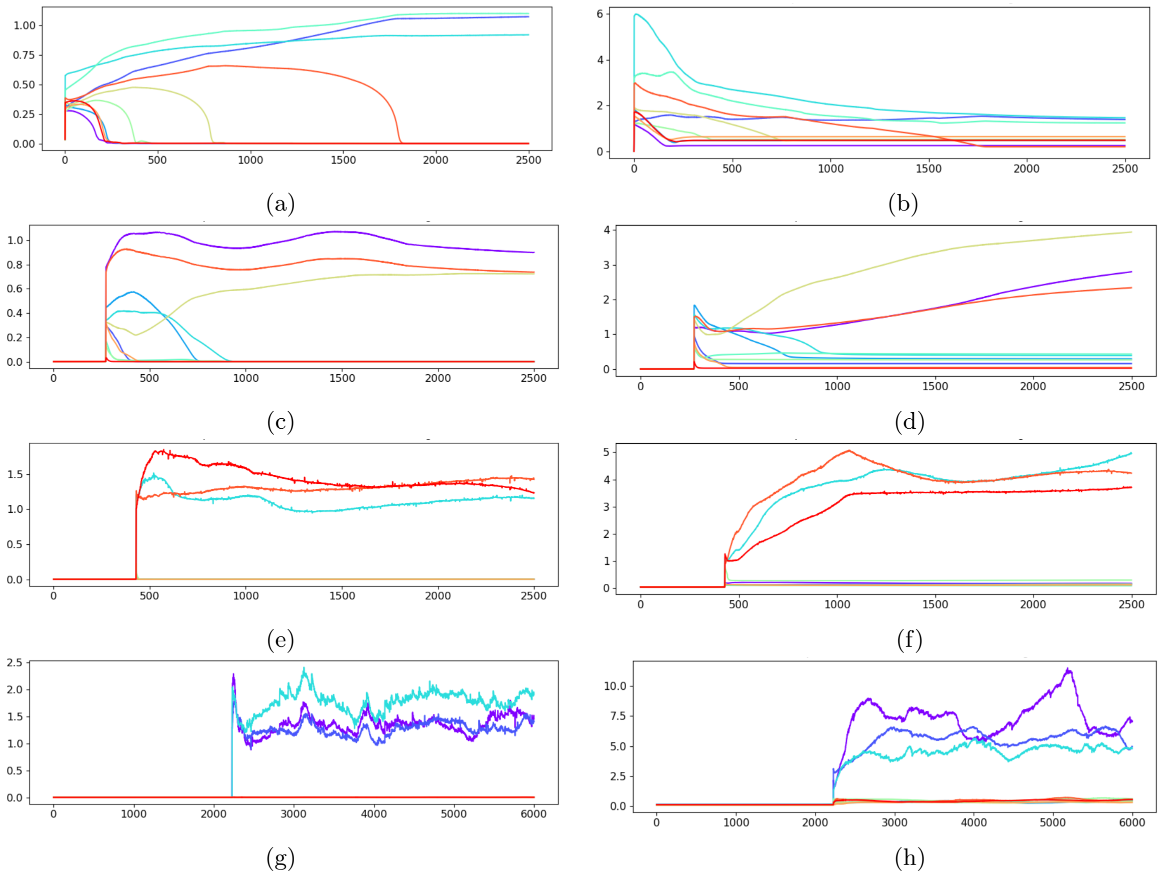

Figure 3: Evolution of the squared norms of the core slices $||\mathcal{G}_{i,:,:}(t)||^2$ (right column) and overparameterized vectors $||\prod A_k^{1,i}(t)\omega_i^1(t)||$ (left column) of deep Tucker factorization for mode 1 during gradient descent iterations. Rows 1 to 4 correspond to model depth from 1 to 4, respectively.

could also describe implicit regularization towards low tensor rank. The following proposition shows that the structured sparsity of the core tensor $\mathcal{G}$ promotes low multilinear rank solutions.

Proposition 4.6. *Let* $\mathcal{W}_e = \mathcal{G} \times_1 V^{(1)} \times_2 V^{(2)} \cdots \times_N V^{(N)}$, *where* $V^{(n)} = \left[\prod_{i=1}^{k_n} A_i^{n,1}\omega_1^n, \dots, \prod_{i=1}^{k_n} A_i^{n,d_n}\omega_{d_n}^n\right]$, *the deep Tucker factorization learned with gradient descent. Assume that* $\|V^{(n)}\| \neq 0$, $\forall n \in [\![1,N]\!]$*. For any* $\epsilon > 0$ *and* $n \in [\![1,N]\!]$*, let* R_n *be the number of mode-n subtensors of* $\mathcal{G}$ *satisfying* $\|\mathcal{G}_{:\dots:r_n:\dots:}\| > \dfrac{\epsilon}{\sqrt{d_n N}\prod_{n=1}^N \|V^{(n)}\|_2}$, *then we have*

$$\inf_{\substack{\mathcal{W}\in\mathbb{R}^{d_1\times\cdots\times d_N}\\ \mathrm{multirank}(\mathcal{W})\le(\mathrm{R_1},\dots,\mathrm{R_N})}} \|\mathcal{W}_e-\mathcal{W}\| \le \epsilon. \tag{4}$$

Note that R_n in Proposition 4.6 becomes smaller when the number of subtensors of the core tensor $\mathcal{G}$ with near zero-norm increases. When the depth is large, this effect is more pronounced. This means that we are able to well-approximate the solution by a tensor with low multilinear rank. To get some intuition, by applying mode-n matricization to (1), we obtain

$$[\mathcal{W}]_{(n)} = V^{(n)}[\mathcal{G}]_{(n)}\Big(V^{(N)}\odot\dots \odot V^{(n+1)}\odot V^{(n-1)}\odot\dots V^{(1)}\Big)^\top,$$

where $[\mathcal{G}]_{(n)}$ is the matricization of $\mathcal{G}$ in the mode n. The number of non-zero rows of $[\mathcal{G}]_{(n)}$ is equal to the number of subtensors of $\mathcal{G}$ in the mode n with non-zero norm, which is small at the end of the optimization because of the implicit structured sparsity regularization. Thus $[\mathcal{G}]_{(n)}$ is a low-rank matrix. Since $rank([\mathcal{W}]_{(n)}) \leq rank([\mathcal{G}]_{(n)})$, then $[\mathcal{W}]_{(n)}$ is also of low-rank for every mode n, and by consequence $\mathcal{W}$ has a low multilinear rank.

5 NUMERICAL EXPERIMENTS

We present and discuss numerical results on synthetic data to illustrate our theoretical findings. We analyze the training of deep Tucker models to observe

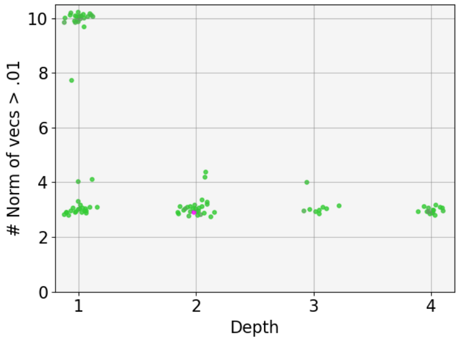

Figure 4: Effect of the depth on sparsity and generalization error. This figure shows the number of overparameterized vectors from mode 1 with squared norm greater than 0.01 at the end of training. Colors corresponds to generalization performance, lighter green the better, ranging from 10^{-12} to 10^{-4} MSE. Deeper models lead to sparse and good solutions.

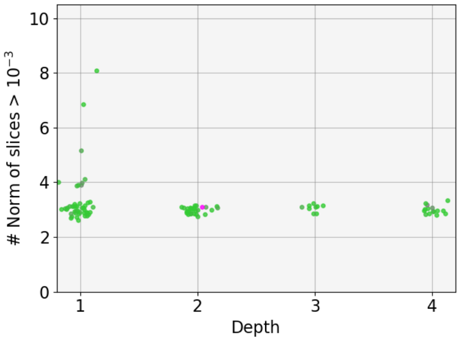

Figure 5: Effect of the depth on the rank. The figure shows the number of slices of the tensor core in mode 1 with norm greater than 10^{-4} at the end of training after HOSVD (without rank constraint). Colors corresponds to generalization performance, ranging from 10^{-12} to 10^{-4} MSE, lighter green the better. Deeper models lead to low-rank and good solutions.

how it may lead to an implicit regularization towards structured sparsity.

Synthetic data We consider a $(10, 10, 10)$ random tensor of Tucker multirank $(3, 3, 3)$, built such that each slice of its Tucker core has a norm less than 1. The tensor is low-rank but not sparse. We train various completion models from shallow to deep with 70% of observed values; the goal is to complete the 30% missing values. We used Tensorflow toolbox (Abadi et al., 2015) and optimized our models with Adam and a learning rate in $\{0.001, \ldots, 0.005\}$. In the set of experiments below, we initialized the parameters from a normal distribution with zero mean and standard deviation in $\{0.0005, .., 0.005\}$.

Effect of depth on structured sparsity Figure 3 depicts the norms of deep Tucker parameters over iterations during the training for various depths from 1 (3a and 3b) to 4 (3g and 3h). The left column shows the evolution of $||\prod_{k=1}^{K} A_k^{1,i}(t)\omega_i^1(t)||$ where $K+1$ is the depth of the model. The right column shows the squared norm of the corresponding slices of the tucker core $||\mathcal{G}_{i,:,:}(t)||^2$ through iterations. Rows 1 to 4 correspond to model depth from 1 to 4, respectively.

We observe that structured sparsity is favored by deeper models. For depth 1 and depth 2, some of the slices emerge then ultimately drop closer to zero. With more overparameterization (i.e., depth 3 to 4), sparsity is more pronounced and is reached much faster.

We now focus on the generalization performance of solutions found by deep tucker factorization. We consider various models from depth 1 to 4 with the same initialization used for Figure 3 and various seeds, ignoring runs that does not converge after our max iteration budget. Every reported model achieves an error less than 10^{-6}.

Figure 4 shows the number of overparameterized vectors $\prod A_k^{1,i}\omega_i^1$ with a norm greater than 0.01 , at the end of training , for many models varying in depth, initialization and optimization settings. The color of each point depicts test error at convergence from green (lowest error) to brown (highest error). We see that deeper models converged to sparse solutions while achieving low test error. Figure 5 reports the Tucker rank in the mode 1 after performing HOSVD (without rank constraint) on the solution. We can see that the implicit regularization towards structured sparsity in the model parameters induces an implicit regularization towards low Tucker rank.

Real data and baseline comparison We performed experiments using real-world data. We used IL-2[1] and COVID[2] datasets, which contain mutein treatment responses and COVID-19 systems serology, resulting in tensors of dimensions $(11, 4, 12)$ and $(10, 6, 11)$, respectively. The results are depicted in Figures 1 and 2, which show completion performance with 70% of missing data for multiple runs with various initializations and learning rates. Every single point stands for an experiment. Points are plotted with a small random displacement in x and y coordinates to better see point clouds. Colors correspond to R2-score, lighter green the better, ranging from 0.94 to 0.97 and 0.8 to 0.9 for

[1] http://tensorly.org/stable/modules/generated/tensorly.datasets.load_IL2data.html

[2] http://tensorly.org/stable/modules/generated/tensorly.datasets.load_covid19_serology.html

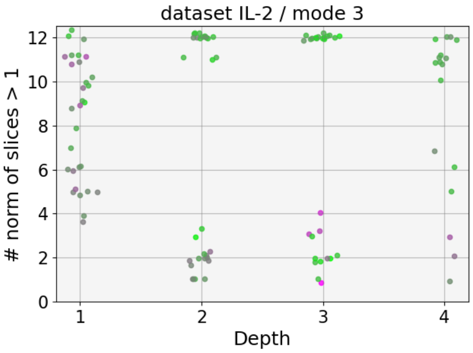

Figure 6: Effect of the depth on sparsity and generalization error on IL-2 dataset. Colors corresponds to R2-score performance, lighter-green the better, ranging from 0.94 to 0.97.

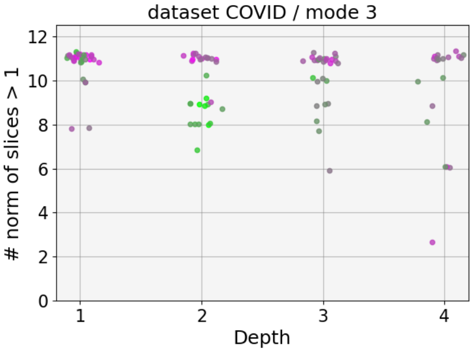

Figure 7: Effect of the depth on sparsity and generalization error on COVID dataset. Colors corresponds to R2-score performance, lighter-green the better, ranging from 0.8 to 0.9.

Methods	Rank	R2
IL-2 dataset		
HOOI	(3,4,3)	0.909
DeepTucker (depth 2)		**0.968**
HOOI	(3,4,2)	0.925
DeepTucker (depth 3)		**0.971**
COVID-19 dataset		
HOOI	(9,6,9)	**0.899**
DeepTucker (depth 2)		0.869
HOOI	(8,6,8)	0.625
DeepTucker (depth 4)		**0.867**

Table 1: Baseline comparison on the two real-world datasets. We report the R2-score of tensor completion with deep Tucker factorization and HOOI.

IL-2 and COVID datasets respectively. These experimental results corroborate the simulations and confirm our theoretical findings. In Table 1, we also compare R2-score on both datasets with Tucker-HOOI (Kolda and Bader, 2009)(available in Tensorly) using the multirank for which our model converged. We observe a better performance for our method, notably for deeper models.

6 DISCUSSION

This work is the first to our knowledge to theoretically characterize implicit regularization in Deep Tucker factorization. Compared to previous works, which considered matrix rank or tensor CP rank (Arora et al., 2019; Razin et al., 2021), we focused on multilinear rank which is based on the Tucker decomposition, a popular technique for many data analysis and ML applications (Cichocki et al., 2016). In addition, we have shown an implicit regularization towards structured sparsity. Such an implicit regularization was observed recently in Li et al. (2023) in the setting of linear neural networks. Our study extends this observation to include deep nonlinear neural networks (as depicted in Figure 2, deep Tucker decomposition corresponds to a certain types of nonlinear neural networks). It is worth noting that Tucker decomposition is a generalization of CP decomposition (the core tensor of CP decomposition is diagonal), and the dynamical analysis in this case is more challenging and adds technical difficulties which we handle by showing time-invariant dynamic pattern between the core tensor, the weight vectors and the depth matrices.

There are differences between our approach and that of Razin et al. (2022) about the implicit bias over hierarchical tensor (HT) factorization. The notion of depth is different. In HT, depth is associated with the hierarchy of factorization and it wasn't shown if it has an effect on the implicit regularization. However, we relied on deep matrix factorization to define a notion of depth for Tucker factorization that enhances implicit bias. The hierarchical structure of HT imposes a notion of locality and so the attention was towards implicit regularization of local components of the tensor. In deep Tucker model, we don't have this notion of locality; instead, we have implicit regularization that enhances structured sparsity.

7 CONCLUSION

We provided a theoretical analysis of the dynamics of gradient flow over Tucker tensor factorization. Our analysis provides insight about the mechanism of implicit regularization in deep learning. It shows that deep Tucker factorization trained by gradient descent could induce a structured sparse regularization, which can have the effect of biasing the gradient descent towards solutions with low multilinear rank.

Acknowledgements

Research reported in this paper was partially supported by PHC Utique no. 44318NJ granted by the Ministry of Higher Education and Scientific Research of Tunisia and the Ministry of Foreign Affairs in France.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In *International Conference on Machine Learning*, pages 244–253, 2018.

Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. *Advances in Neural Information Processing Systems*, 32, 2019.

Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116 (32):15849–15854, 2019.

Lenaic Chizat and Francis Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Conference on Learning Theory*, pages 1305–1338, 2020.

Andrzej Cichocki, Namgil Lee, Ivan Oseledets, Anh-Huy Phan, Qibin Zhao, Danilo P Mandic, et al. Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions. *Foundations and Trends® in Machine Learning*, 9(4-5):249–429, 2016.

Nadav Cohen, Or Sharir, and Amnon Shashua. On the expressive power of deep learning: A tensor analysis. In *Conference on learning theory*, pages 698–728, 2016.

Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

Rong Ge, Yunwei Ren, Xiang Wang, and Mo Zhou. Understanding deflation process in over-parametrized tensor decomposition. *Advances in Neural Information Processing Systems*, 34:1299–1311, 2021.

Daniel Gissin, Shai Shalev-Shwartz, and Amit Daniely. The implicit bias of depth: How incremental learning drives generalization. In *International Conference on Learning representations*, 2020.

Suriya Gunasekar, Blake E Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Implicit regularization in matrix factorization. *Advances in Neural Information Processing Systems*, 30, 2017.

Kais Hariz, Hachem Kadri, Stéphane Ayache, Maher Moakher, and Thierry Artières. Implicit regularization with polynomial growth in deep tensor factorization. In *International Conference on Machine Learning*, pages 8484–8501, 2022.

Bo Jiang, Fan Yang, and Shuzhong Zhang. Tensor and its tucker core: the invariance relationships. *Numerical Linear Algebra with Applications*, 24(3):e2086, 2017.

Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

Jiangyuan Li, Thanh Nguyen, Chinmay Hegde, and Ka Wai Wong. Implicit sparse regularization: The impact of depth and early stopping. *Advances in Neural Information Processing Systems*, 34:28298–28309, 2021a.

Jiangyuan Li, Thanh V Nguyen, Chinmay Hegde, and Raymond KW Wong. Implicit regularization for group sparsity. In *International Conference on Learning representations*, 2023.

Zhiyuan Li, Yuping Luo, and Kaifeng Lyu. Towards resolving the implicit bias of gradient descent for matrix factorization: Greedy low-rank learning. In *International Conference on Learning representations*, 2021b.

Paolo Milanesi, Hachem Kadri, Stéphane Ayache, and Thierry Artières. Implicit regularization in deep tensor factorization. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2021.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

Noam Razin and Nadav Cohen. Implicit regularization in deep learning may not be explainable by norms. *Advances in neural information processing systems*, 33:21174–21187, 2020.

Noam Razin, Asaf Maman, and Nadav Cohen. Implicit regularization in tensor factorization. In *International Conference on Machine Learning*, pages 8913–8924, 2021.

Noam Razin, Asaf Maman, and Nadav Cohen. Implicit regularization in hierarchical tensor factorization and deep convolutional neural networks. In *International Conference on Machine Learning*, pages 18422–18462, 2022.

Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

Liu Ziyin. Symmetry leads to structured constraint of learning. *arXiv preprint arXiv:2309.16932*, 2023.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Not Applicable]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Not Applicable]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [No] We used our internal cluster composed of 30+ Nvidia V100 GPUs.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A Proofs

First, let us recall that $\mathcal{A} \in \mathbb{R}^{d_1 \times \ldots \times d_N}$ is the ground truth tensor that we want to recover and $\Omega \subset [\![1, d_1]\!] \times \ldots \times [\![1, d_N]\!]$ is the set of indexes of observed elements. We minimize $\mathcal{L}(\mathcal{W}) = \frac{1}{|\Omega|} \sum_{(i_1,\ldots,i_N)\in\Omega} \ell(\mathcal{W}_{i_1,\ldots,i_N} - \mathcal{A}_{i_1,\ldots,i_N})$ with $\mathcal{W} \in \mathbb{R}^{d_1 \times \ldots \times d_N}$ having the following deep Tucker factorization:

$$\mathcal{W} = \mathcal{G} \times_1 \left(\overbrace{\prod_{i=1}^{k_1} A_i^{1,1} \omega_1^1}^{\text{column } 1}, \ldots, \overbrace{\prod_{i=1}^{k_1} A_i^{1,d_1} \omega_{d_1}^1}^{\text{column } d_1} \right) \times_2 \ldots \times_N \left(\prod_{i=1}^{k_N} A_i^{N,1} \omega_1^N, \ldots, \prod_{i=1}^{k_N} A_i^{N,d_N} \omega_{d_N}^N \right)$$

with $\mathcal{G} \in \mathbb{R}^{d_1 \times \ldots \times d_N}$, $A_i^{n,r_n} \in \mathbb{R}^{d_n \times d_n}$ and $\omega_{r_n}^n \in \mathbb{R}^{d_n}$. $\mathcal{W}$ can also be written:

$$\mathcal{W} = \sum_{r_1=1}^{d_1} \cdots \sum_{r_N=1}^{d_N} \mathcal{G}_{r_1,\ldots,r_N} \bigotimes_{n=1}^{N} \prod_{i=1}^{k_n} A_i^{n,r_n} \omega_{r_n}^n .$$

Consider that $\mathcal{L}(\mathcal{W}) = \Phi\left(\{\omega_{r_n}^n\}_{n=1}^{N}, \{A_i^{n,r_n}\}_{n=1,\ r_n=1,\ i=1}^{N\quad d_n\quad k_n}, \{\mathcal{G}_{r_1,\ldots,r_N=1}\}_{r_1=1,\ldots,r_N}^{d_1,\ldots,d_N} \right)$. Using gradient descent with infinitesimally small learning rate and near zero initialization we have:

$$\frac{d}{dt}\omega_{r_m}^m(t) = -\frac{\partial}{\partial \omega_{r_m}^m} \Phi\left(\{\omega_{r_n}^n(t)\}_{n=1,r_n=1}^{N\quad d_n}, \{A_i^{n,r_n}(t)\}_{n=1,\ r_n=1,\ i=1}^{N\quad d_n\quad k_n}, \{\mathcal{G}_{r_1,\ldots,r_N}(t)\}_{r_1=1,\ldots,r_N=1}^{d_1,\ldots,d_N} \right),$$

$$\frac{d}{dt}A_j^{m,r_m}(t) = -\frac{\partial}{\partial A_j^{m,r_m}} \Phi\left(\{\omega_{r_n}^n(t)\}_{n=1,r_n=1}^{N\quad d_n}, \{A_i^{n,r_n}(t)\}_{n=1,\ r_n=1,\ i=1}^{N\quad d_n\quad k_n}, \{\mathcal{G}_{r_1,\ldots,r_N}(t)\}_{r_1=1,\ldots,r_N=1}^{d_1,\ldots,d_N} \right),$$

$$\frac{d}{dt}\mathcal{G}_{r_1,\ldots,r_N}(t) = -\frac{\partial}{\partial \mathcal{G}_{r_1,\ldots,r_N}} \Phi\left(\{\omega_{r_n}^n(t)\}_{n=1,r_n=1}^{N\quad d_n}, \{A_i^{n,r_n}(t)\}_{n=1,\ r_n=1,\ i=1}^{N\quad d_n\quad k_n}, \{\mathcal{G}_{s_1,\ldots,s_N}(t)\}_{s_1=1,\ldots,s_N=1}^{d_1,\ldots,d_N} \right).$$

A.1 Proof of Lemma 4.1

To show Lemma 4.1, we will use the following result shown in Razin et al. (2021).

Lemma A.1. *$\forall \mathcal{A} \in \mathbb{R}^{d_1 \times \ldots \times d_N}$ and $\{w^n \in \mathbb{R}^{d_n}\}_{n=1}^N$ where $d_1, \ldots d_N \in \mathbb{N}$, it holds that*

$$\left\langle \mathcal{A}, \bigotimes_{n'=1}^{N} w^{n'} \right\rangle = \left\langle [\mathcal{A}]_{(n)} \cdot \bigodot_{n' \neq n} w^{n'}, w^n \right\rangle, \ n = 1, \ldots, N$$

where $[\mathcal{A}]_{(n)}$ is matricization of the tensor $\mathcal{A}$ in the mode n, and $\odot$ is the Kronecker product.

Proof of Lemma 4.1. i. We compute $\frac{d}{dt}\|\omega_{r_m}^m(t)\|^2$. We assume that $\{\omega_{r_n}^n\}_{(n,r_n)\neq(m,r_m)}$, $\{A_i^{n,r_n}\}_{n=1,\ r_n=1,\ i=1}^{N\quad d_n\quad k_n}$ and $\{\mathcal{G}_{r_1,\ldots,r_N}\}_{r_1=1,\ldots,r_N=1}^{d_1,\ldots,d_N}$ are fixed and consider:

$$\Phi_{r_m}^m(\omega_{r_m}^m) = \Phi\left(\{\omega_{r_n}^n\}_{n=1,r_n=1}^{N\quad d_n}, \{A_i^{n,r_n}\}_{n=1,\ r_n=1,\ i=1}^{N\quad d_n\quad k_n}, \{\mathcal{G}_{r_1,\ldots,r_N}\}_{r_1=1,\ldots,r_N=1}^{d_1,\ldots,d_N} \right).$$

For $\Delta \in \mathbb{R}^{d_m}$, using first-order Taylor approximation we have

$$
\begin{aligned}
\Phi^m_{r_m}(\omega^m_{r_m} + \Delta) &= \mathcal{L}\Bigg(\mathcal{W} + \sum_{r_1,\dots,r_{m-1},r_{m+1},\dots,r_N} \mathcal{G}_{r_1,\dots,r_N} \\
&\qquad \bigotimes_{n=1}^{m-1} \prod_{i=1}^{k_n} A_i^{n,r_n}\omega^n_{r_n} \otimes \prod_{i=1}^{k_m} A_i^{m,r_m}\Delta \otimes \bigotimes_{n=m+1}^{N} \prod_{i=1}^{k_n} A_i^{n,r_n}\omega^n_{r_n}\Bigg) \\
&= \mathcal{L}(\mathcal{W}) + \Bigg\langle \nabla\mathcal{L}(\mathcal{W}), \sum_{r_1,\dots,r_{m-1},r_{m+1},\dots,r_N} \mathcal{G}_{r_1,\dots,r_N} \\
&\qquad \bigotimes_{n=1}^{m-1} \prod_{i=1}^{k_n} A_i^{n,r_n}\omega^n_{r_n} \otimes \prod_{i=1}^{k_m} A_i^{m,r_m}\Delta \otimes \bigotimes_{n=m+1}^{N} \prod_{i=1}^{k_n} A_i^{n,r_n}\omega^n_{r_n}\Bigg)\Bigg\rangle + o(\|\Delta\|) \\
&= \mathcal{L}(\mathcal{W}) + \sum_{r_1,\dots,r_{m-1},r_{m+1},\dots,r_N} \mathcal{G}_{r_1,\dots,r_N} \Bigg\langle [\nabla\mathcal{L}(\mathcal{W})]_{(m)} \left(\bigodot_{n\neq m} \prod_{i=1}^{k_n} A_i^{n,r_n}\omega^n_{r_n}\right), \\
&\qquad \prod_{i=1}^{k_m} A_i^{m,r_m}\Delta\Bigg\rangle + o(\|\Delta\|) \\
&= \mathcal{L}(\mathcal{W}) + \Bigg\langle \sum_{r_1,\dots,r_{m-1},r_{m+1},\dots,r_N} \mathcal{G}_{r_1,\dots,r_N} \left(\prod_{i=1}^{k_m} A_i^{m,r_m}\right)^T [\nabla\mathcal{L}(\mathcal{W})]_{(m)} \\
&\qquad \left(\bigodot_{n\neq m} \prod_{i=1}^{k_n} A_i^{n,r_n}\omega^n_{r_n}\right), \Delta\Bigg\rangle + o(\|\Delta\|).
\end{aligned}
$$

Since

$$
\frac{d}{dt}\omega^m_{r_m}(t) = -\frac{\partial\Phi}{\partial\omega^m_{r_m}}\left(\left\{\omega^n_{r_n}(t)\right\}^{N \,\,\, d_n}_{n=1, r_n=1}, \{A_i^{n,r_n}(t)\}^{N \,\,\, d_n \,\,\, k_n}_{n=1, \, r_n=1, \, i=1}, \{\mathcal{G}_{r_1,\dots,r_N}(t)\}^{d_1,\dots,d_N}_{r_1=1,\dots,r_N=1}\right),
$$

then

$$
\frac{d}{dt}\omega^m_{r_m}(t) = -\sum_{r_1,\dots,r_{m-1},r_{m+1},\dots,r_N} \mathcal{G}_{r_1,\dots,r_N}(t)\left(\prod_{i=1}^{k_m} A_i^{m,r_m}(t)\right)^T [\nabla\mathcal{L}(\mathcal{W}(t))]_{(m)} \left(\bigodot_{n\neq m} \prod_{i=1}^{k_n} A_i^{n,r_n}(t)\omega^n_{r_n}(t)\right).
$$

So we have

$$
\begin{aligned}
&\frac{d}{dt}\|\omega^m_{r_m}(t)\|^2 = 2\left\langle \omega^m_{r_m}(t), \frac{d}{dt}\omega^m_{r_m}(t)\right\rangle \\
&= -2\Bigg\langle \omega^m_{r_m}(t), \sum_{r_1,\dots,r_{m-1},r_{m+1},\dots,r_N} \mathcal{G}_{r_1,\dots,r_N}(t)\left(\prod_{i=1}^{k_m} A_i^{m,r_m}(t)\right)^T [\nabla\mathcal{L}(\mathcal{W}(t))]_{(m)} \\
&\qquad\qquad \left(\bigodot_{n\neq m} \prod_{i=1}^{k_n} A_i^{n,r_n}(t)\omega^n_{r_n}(t)\right)\Bigg\rangle \\
&= -2\sum_{r_1,\dots,r_{m-1},r_{m+1},\dots,r_N} \mathcal{G}_{r_1,\dots,r_N}(t)\Bigg\langle \prod_{i=1}^{k_m} A_i^{m,r_m}(t)\omega^m_{r_m}(t), \\
&\qquad\qquad [\nabla\mathcal{L}(\mathcal{W}(t))]_{(m)} \left(\bigodot_{n\neq m} \prod_{i=1}^{k_n} A_i^{n,r_n}(t)\omega^n_{r_n}(t)\right)\Bigg\rangle \\
&= 2\sum_{r_1,\dots,r_{m-1},r_{m+1},\dots,r_N} \mathcal{G}_{r_1,\dots,r_N}(t)\left\langle -\nabla\mathcal{L}(\mathcal{W}(t)), \bigotimes_{n=1}^{N} \prod_{i=1}^{k_n} A_i^{n,r_n}(t)\omega^n_{r_n}(t))\right\rangle. \qquad (5)
\end{aligned}
$$

We now compute $\frac{d}{dt}\mathcal{G}_{r_1,\dots,r_N}(t)$. We assume that $\{\omega^n_{r_n}\}_{n=1}^{N}{}_{r_n=1}^{d_n}$, $\{A_i^{n,r_n}\}_{n=1,}^{N}{}_{r_n=1,}^{d_n}{}_{i=1}^{k_n}$ and $\{\mathcal{G}_{s_1,\dots,s_N}\}_{(s_1,\dots,s_N)\neq(r_1,\dots,r_N)}$ are fixed and consider:

$$\Phi_{r_1,\dots,r_N}(\mathcal{G}_{r_1,\dots,r_N}) = \Phi\left(\{\omega^n_{r_n}\}_{n=1,r_n=1}^{N\ \ d_n}, \{A_i^{n,r_n}\}_{n=1,\ r_n=1,\ i=1}^{N\ \ \ d_n\ \ \ k_n}, \{\mathcal{G}_{s_1,\dots,s_N}\}_{s_1=1,\dots,s_N=1}^{d_1,\dots,d_N}\right).$$

For $\Delta\in\mathbb{R}$, using first-order Taylor approximation we have

$$\begin{aligned}\Phi_{r_1,\dots,r_N}(\mathcal{G}_{r_1,\dots,r_N}+\Delta) &= \mathcal{L}\left(\mathcal{W}+\Delta\bigotimes_{n=1}^{N}\prod_{i=1}^{k_n}A_i^{n,r_n}\omega^n_{r_n}\right)\\ &= \mathcal{L}(\mathcal{W}) + \left\langle\nabla\mathcal{L}(\mathcal{W}), \Delta\bigotimes_{n=1}^{N}\prod_{i=1}^{k_n}A_i^{n,r_n}\omega^n_{r_n}\right\rangle + o(\|\Delta\|)\\ &= \mathcal{L}(\mathcal{W}) + \Delta\left\langle\nabla\mathcal{L}(\mathcal{W}), \bigotimes_{n=1}^{N}\prod_{i=1}^{k_n}A_i^{n,r_n}\omega^n_{r_n}\right\rangle + o(\|\Delta\|)\end{aligned}$$

Since

$$\frac{d}{dt}\mathcal{G}_{r_1,\dots,r_N}(t) = -\frac{\partial}{\partial\mathcal{G}_{r_1,\dots,r_N}}\Phi\left(\left\{\omega^n_{r_n}(t)\right\}_{n=1,r_n=1}^{N\ \ d_n}, \{A_i^{n,r_n}(t)\}_{n=1,\ \ r_n=1,\ \ \ i=1}^{N\ \ \ d_n\ \ \ \ k_n}, \{\mathcal{G}_{s_1,\dots,s_N}(t)\}_{s_1=1,\dots,s_N=1}^{d_1,\dots,d_N}\right),$$

then

$$\frac{d}{dt}\mathcal{G}_{r_1,\dots,r_N}(t) = \left\langle -\nabla\mathcal{L}(\mathcal{W}(t)), \bigotimes_{n=1}^{N}\prod_{i=1}^{k_n}A_i^{n,r_n}(t)\omega^n_{r_n}(t)\right\rangle.$$

So we have

$$\begin{aligned}\frac{d}{dt}\|\omega^m_{r_m}(t)\|^2 &= 2\sum_{r_1,\dots,r_{m-1},r_{m+1},\dots,r_N}\mathcal{G}_{r_1,\dots,r_N}(t)\frac{d}{dt}\mathcal{G}_{r_1,\dots,r_N}(t)\\ &= \sum_{r_1,\dots,r_{m-1},r_{m+1},\dots,r_N}\frac{d}{dt}(\mathcal{G}^2_{r_1,\dots,r_N}(t))\\ &= \frac{d}{dt}\sum_{r_1,\dots,r_{m-1},r_{m+1},\dots,r_N}(\mathcal{G}^2_{r_1,\dots,r_N}(t))\\ &= \frac{d}{dt}\|\mathcal{G}_{:\dots:r_m:\dots:}(t)\|^2,\end{aligned}$$

where $\mathcal{G}_{:\dots:r_m:\dots:}\in\mathbb{R}^{d_1\times\dots\times d_{m-1}\times d_{m+1}\times\dots\times d_N}$ is the subtensor obtained by fixing the index of the mode m equal to r_m.

Thus, $\forall r_m = 1\dots d_m$,

$$\|\omega^m_{r_m}(t)\|^2 - \|G_{:\dots:,r_m,:\dots:}(t)\|^2 = \|\omega^m_{r_m}(0)\|^2 - \|G_{:\dots:,r_m,:\dots:}(0)\|^2.$$

ii. We compute $\frac{d}{dt}\left(\|A_i^{n,r}(t)\|^2\right)$. Assume that $\{A_j^{n,r_n}\}_{(n,r_n,j)\neq(m,r_m,i)}$, $\{\mathcal{G}_{r_1,\dots,r_N}\}_{r_1=1,\dots,r_N=1}^{d_1,\dots,d_N}$ and $\{\omega^n_{r_n}\}_{n=1}^{N}{}_{r_n=1}^{d_n}$

are fixed.

$$
\begin{aligned}
\Phi_i^{m,r_m}(A_i^{m,r_m}+\Delta) &= \mathcal{L}\Bigg(\mathcal{W} + \sum_{r_1,\ldots,r_{m-1},r_{m+1},\ldots,r_N} \mathcal{G}_{r_1,\ldots,r_N} \\
&\quad \bigotimes_{n=1}^{m-1}\prod_{j=1}^{k_n} A_j^{n,r_n}\omega_{r_n}^n \otimes \prod_{j=1}^{i-1} A_j^{m,r_m}\Delta \prod_{j=i+1}^{k_m} A_j^{m,r_m}\omega_{r_m}^m \otimes \bigotimes_{n=m+1}^{N}\prod_{j=1}^{k_n} A_j^{n,r_n}\omega_{r_n}^n\Bigg) + o(\|\Delta\|) \\
&= \mathcal{L}(\mathcal{W}) + \sum_{r_1,\ldots,r_{m-1},r_{m+1},\ldots,r_N} \mathcal{G}_{r_1,\ldots,r_N}\Bigg\langle \nabla\mathcal{L}(\mathcal{W}), \\
&\quad \bigotimes_{n=1}^{m-1}\prod_{j=1}^{k_n} A_j^{n,r_n}\omega_{r_n}^n \otimes \prod_{j=1}^{i-1} A_j^{m,r_m}\Delta \prod_{j=i+1}^{k_m} A_j^{m,r_m}\omega_{r_m}^m \otimes \bigotimes_{n=m+1}^{N}\prod_{j=1}^{k_n} A_j^{n,r_n}\omega_{r_n}^n\Bigg\rangle + o(\|\Delta\|) \\
&= \mathcal{L}(\mathcal{W}) + \sum_{r_1,\ldots,r_{m-1},r_{m+1},\ldots,r_N} \mathcal{G}_{r_1,\ldots,r_N}\Bigg\langle [\nabla\mathcal{L}(\mathcal{W})]_{(m)}\left(\bigodot_{n\neq m}\prod_{j=1}^{k_n} A_j^{n,r_n}\omega_{r_n}^n\right), \\
&\quad \prod_{j=1}^{i-1} A_j^{m,r_m}\Delta \prod_{j=i+1}^{k_m} A_j^{m,r_m}\omega_{r_m}^m\Bigg\rangle + o(\|\Delta\|) \\
&= \mathcal{L}(\mathcal{W}) + \sum_{r_1,\ldots,r_{m-1},r_{m+1},\ldots,r_N} \mathcal{G}_{r_1,\ldots,r_N}\Bigg\langle \left(\prod_{j=1}^{i-1} A_j^{m,r_m}\right)^T [\nabla\mathcal{L}(\mathcal{W})]_{(m)}\left(\bigodot_{n\neq m}\prod_{j=1}^{k_n} A_j^{n,r_n}\omega_{r_n}^n\right) \\
&\quad \left(\prod_{j=i+1}^{k_m} A_j^{m,r_m}\omega_{r_m}^m\right)^T, \Delta\Bigg\rangle + o(\|\Delta\|) \\
&= \mathcal{L}(\mathcal{W}) + \Bigg\langle \sum_{r_1,\ldots,r_{m-1},r_{m+1},\ldots,r_N} \mathcal{G}_{r_1,\ldots,r_N}\left(\prod_{j=1}^{i-1} A_j^{m,r_m}\right)^T [\nabla\mathcal{L}(\mathcal{W})]_{(m)}\left(\bigodot_{n\neq m}\prod_{j=1}^{k_n} A_j^{n,r_n}\omega_{r_n}^n\right) \\
&\quad {\omega_{r_m}^m}^T\left(\prod_{j=i+1}^{k_m} A_j^{m,r_m}\right)^T, \Delta\Bigg\rangle + o(\|\Delta\|).
\end{aligned}
$$

Since

$$
\frac{d}{dt}A_j^{m,r_m}(t) = -\frac{\partial}{\partial A_j^{m,r_m}}\Phi\left(\left\{\omega_{r_n}^n(t)\right\}_{n=1,r_n=1}^{N\quad d_n}, \{A_i^{n,r_n}(t)\}_{n=1,\ r_n=1,\ i=1}^{N\quad d_n\quad k_n}, \{\mathcal{G}_{r_1,\ldots,r_N}(t)\}_{r_1=1,\ldots,r_N=1}^{d_1,\ldots,d_N}\right),
$$

then

$$
\begin{aligned}
\frac{d}{dt}A_i^{m,r_m}(t) = -\sum_{r_1,\ldots,r_{m-1},r_{m+1},\ldots,r_N} \mathcal{G}_{r_1,\ldots,r_N}(t)\left(\prod_{j=1}^{i-1} A_j^{m,r_m}(t)\right)^T [\nabla\mathcal{L}(\mathcal{W}(t))]_{(m)} \\
\left(\bigodot_{n\neq m}\prod_{j=1}^{k_n} A_j^{n,r_n}(t)\omega_{r_n}^n(t)\right)\omega_{r_m}^m(t)^T\left(\prod_{j=i+1}^{k_m} A_j^{m,r_m}(t)\right)^T.
\end{aligned}
\tag{6}
$$

So we have

$$\begin{aligned}
&\frac{d}{dt}\|A_i^{m,r_m}(t)\|^2 = 2\left\langle A_i^{m,r_m}(t), \frac{d}{dt}A_i^{m,r_m}(t)\right\rangle \\
&= 2\Bigg\langle A_i^{m,r_m}(t), -\sum_{r_1,\dots,r_{m-1},r_{m+1},\dots,r_N} G_{r_1,\dots,r_N}(t)\left(\prod_{j=1}^{i-1} A_j^{m,r_m}(t)\right)^T [\nabla\mathcal{L}(\mathcal{W}(t))]_{(m)} \\
&\qquad\qquad \left(\bigodot_{n\neq m}\prod_{j=1}^{k_n} A_j^{n,r_n}(t)\omega_{r_n}^n(t)\right)\omega_{r_m}^m(t)^T\left(\prod_{j=i+1}^{k_m} A_j^{m,r_m}(t)\right)^T\Bigg\rangle \\
&= -2\sum_{r_1,\dots,r_{m-1},r_{m+1},\dots,r_N} G_{r_1,\dots,r_N}(t)\Bigg\langle\prod_{j=1}^{i-1} A_j^{m,r_m}(t)A_i^{m,r_m}(t)\prod_{j=i+1}^{k_m} A_j^{m,r_m}(t)\omega_{r_m}^m(t), [\nabla\mathcal{L}(\mathcal{W}(t))]_{(m)} \\
&\qquad\qquad \left(\bigodot_{n\neq m}\prod_{j=1}^{k_n} A_j^{n,r_n}(t)\omega_{r_n}^n(t)\right)\Bigg\rangle \\
&= 2\sum_{r_1,\dots,r_{m-1},r_{m+1},\dots,r_N} G_{r_1,\dots,r_N}(t)\left\langle -\nabla\mathcal{L}(\mathcal{W}(t)), \bigotimes_{n=1}^{N}\prod_{j=1}^{k_n} A_j^{n,r_n}(t)\omega_{r_n}^n(t)\right\rangle.
\end{aligned}$$

Then, $\frac{d}{dt}\|\omega_{r_m}^m(t)\|^2 = \frac{d}{dt}\|A_i^{m,r_m}(t)\|^2,\ \forall i = 1,\dots,k_m$, and thus, $\forall i = 1,\dots,k_m,\ \forall t \geq 0$,

$$\|A_i^{m,r_m}(t)\|^2 - \|\omega_{r_m}^m(t)\|^2 = \|A_i^{m,r_m}(0)\|^2 - \|\omega_{r_m}^m(0)\|^2 \tag{7}$$

iii. Subtracting the same equation (7) with i replaced by j, we obtain, $\forall i,j = 1,\dots,k_m,\ \forall t \geq 0$,

$$\|A_i^{m,r_m}(t)\|^2 - \|A_j^{m,r_m}(t)\|^2 = \|A_i^{m,r_m}(0)\|^2 - \|A_j^{m,r_m}(0)\|^2.$$

□

A.2 Proof of Theorem 4.3

Proof. Recall that $\varepsilon_1(t) = \max_{\substack{m=1\dots N\\ r_m=1\dots d_m}} \left|\|\omega_{r_m}^m(t)\|^2 - \|\mathcal{G}_{:\dots:r_m:\dots:}(t)\|^2\right|$. If $\varepsilon(0) = 0$, then $\varepsilon_1(0) = 0$. Moreover, by Lemma 4.1 (i), we have $\varepsilon_1(t)$ is constant over time. Then $\varepsilon_1(t) = 0,\ \forall t \geq 0$, which implies that; $\forall t \geq 0$

$$\|\omega_{r_m}^m(t)\| = \|\mathcal{G}_{:\dots:r_m:\dots:}(t)\|. \tag{8}$$

Then

$$
\begin{aligned}
&\frac{d}{dt}\|\mathcal{G}_{:\dots:r_m:\dots:}(t)\| = \frac{d}{dt}\|\omega^m_{r_m}(t)\| = \frac{1}{2}\frac{1}{\|\omega^m_{r_m}(t)\|}\frac{d}{dt}\|\omega^m_{r_m}(t)\|^2 \\
&= \frac{1}{\|\omega^m_{r_m}(t)\|}\sum_{r_1,\dots,r_{m-1},r_{m+1},\dots,r_N} \mathcal{G}_{r_1,\dots,r_N}(t)\left\langle -\nabla\mathcal{L}(\mathcal{W}(t)), \bigotimes_{n=1}^{N}\prod_{j=1}^{k_n} A_j^{n,r_n}(t)\omega^n_{r_n}(t)\right\rangle \text{(using (5))} \\
&= \frac{1}{\|\omega^m_{r_m}(t)\|}\sum_{r_1,\dots,r_{m-1},r_{m+1},\dots,r_N} \mathcal{G}_{r_1,\dots,r_N}(t)\prod_{n=1}^{N}\left\|\prod_{j=1}^{k_n} A_j^{n,r_n}(t)\omega^n_{r_n}(t)\right\| \\
&\qquad\qquad\left\langle -\nabla\mathcal{L}(\mathcal{W}(t)), \bigotimes_{n=1}^{N}\frac{\prod_{j=1}^{k_n} A_j^{n,r_n}(t)\omega^n_{r_n}(t)}{\left\|\prod_{j=1}^{k_n} A_j^{n,r_n}(t)\omega^n_{r_n}(t)\right\|}\right\rangle \\
&= \frac{\left\|\prod_{j=1}^{k_m} A_j^{m,r_m}(t)\omega^m_{r_m}(t)\right\|}{\|\omega^m_{r_m}(t)\|}\sum_{r_1,\dots,r_{m-1},r_{m+1},\dots,r_N} \mathcal{G}_{r_1,\dots,r_N}(t)\left(\prod_{n\neq m}\left\|\prod_{j=1}^{k_n} A_j^{n,r_n}(t)\omega^n_{r_n}(t)\right\|\right) \\
&\qquad\qquad\left\langle -\nabla\mathcal{L}(\mathcal{W}(t)), \bigotimes_{n=1}^{N}\frac{\prod_{j=1}^{k_n} A_j^{n,r_n}(t)\omega^n_{r_n}(t)}{\left\|\prod_{j=1}^{k_n} A_j^{n,r_n}(t)\omega^n_{r_n}(t)\right\|}\right\rangle .
\end{aligned}
$$

□

A.3 Proof of Corollary 4.4

Proof. First, we use the same arguments of the proof of Theorem 1 in Arora et al. (2018) to prove that:

$$
A_i^{m,r_m}(t)^\top A_i^{m,r_m}(t) = A_{i+1}^{m,r_m}(t)A_{i+1}^{m,r_m}(t)^\top \quad ,\ \forall i \in [\![1, k_m - 1]\!], \forall t \geq 0.
$$

Using (6), we have

$$
\begin{aligned}
\frac{d}{dt}A_i^{m,r_m}(t) = &-\left(\prod_{j=1}^{i-1} A_j^{m,r_m}(t)\right)^\top [\nabla\mathcal{L}(\mathcal{W}(t))]_{(m)} \\
&\left(\sum_{r_1,\dots,r_{m-1},r_{m+1},\dots,r_N} \mathcal{G}_{r_1,\dots,r_N}(t)\bigodot_{n\neq m}\prod_{j=1}^{k_n} A_j^{n,r_n}(t)\omega^n_{r_n}(t)\right)\omega^m_{r_m}(t)^\top\left(\prod_{j=i+1}^{k_m} A_j^{m,r_m}(t)\right)^\top .
\end{aligned}
$$

Since

$$
\frac{d}{dt}\left(A_i^{m,r_m}(t)^\top A_i^{m,r_m}(t)\right) = \left(\frac{d}{dt}A_i^{m,r_m}(t)\right)^\top A_i^{m,r_m}(t) + A_i^{m,r_m}(t)^\top \frac{d}{dt}A_i^{m,r_m}(t),
$$

then

$$
\begin{aligned}
&\frac{d}{dt}\left(A_i^{m,r_m}(t)^\top A_i^{m,r_m}(t)\right) = \\
&-\prod_{j=i+1}^{k_m} A_j^{m,r_m}(t)\omega_{r_m}^m(t)\left(\sum_{r_1,\dots,r_{m-1},r_{m+1},\dots,r_N}\mathcal{G}_{r_1,\dots,r_N}(t)\bigodot_{n\neq m}\prod_{j=1}^{k_n} A_j^{n,r_n}(t)\omega_{r_n}^n(t)\right)^\top [\nabla\mathcal{L}(\mathcal{W}(t))]_{(m)}^\top \prod_{j=1}^{i} A_j^{m,r_m}(t) \\
&-\left(\prod_{j=1}^{i} A_j^{m,r_m}(t)\right)^T [\nabla\mathcal{L}(\mathcal{W}(t))]_{(m)}\left(\sum_{r_1,\dots,r_{m-1},r_{m+1},\dots,r_N}\mathcal{G}_{r_1,\dots,r_N}(t)\bigodot_{n\neq m}\prod_{j=1}^{k_n} A_j^{n,r_n}(t)\omega_{r_n}^n(t)\right)\omega_{r_m}^m(t)^\top\left(\prod_{j=i+1}^{k_m} A_j^{m,r_m}(t)\right)^\top .
\end{aligned}
$$

We also compute $\frac{d}{dt}\left(A_{i+1}^{m,r_m}(t)A_{i+1}^{m,r_m}(t)^\top\right)$ using the same way and obtain that

$$
\frac{d}{dt}\left(A_i^{m,r_m}(t)^\top A_i^{m,r_m}(t)\right) = \frac{d}{dt}\left(A_{i+1}^{m,r_m}(t)A_{i+1}^{m,r_m}(t)^\top\right).
$$

Using the fact that $A_i^{m,r_m}(0)^\top A_i^{m,r_m}(0) = A_{i+1}^{m,r_m}(0)A_{i+1}^{m,r_m}(0)^\top$, we then have, $\forall t \geq 0$,

$$
A_i^{m,r_m}(t)^\top A_i^{m,r_m}(t) = A_{i+1}^{m,r_m}(t)A_{i+1}^{m,r_m}(t)^\top. \tag{9}
$$

We now consider singular value decompositions of $A_i^{m,r_m}(t)$ and $A_{i+1}^{m,r_m}(t)$: $A_i^{m,r_m}(t) = U_i^{m,r_m}(t)\Sigma_i^{m,r_m}(t)V_i^{m,r_m}(t)^\top$ and $A_{i+1}^{m,r_m}(t) = U_{i+1}^{m,r_m}(t)\Sigma_{i+1}^{m,r_m}(t)V_{i+1}^{m,r_m}(t)^\top$, where $\Sigma_i^{m,r_m}(t)$ and $\Sigma_{i+1}^{m,r_m}(t)$ are diagonal matrices whose diagonal entries are the singular values of $A_i^{m,r_m}(t)$ and $A_{i+1}^{m,r_m}(t)$, respectively, ordered in decreasing order. Using (9), we obtain

$$
V_i^{m,r_m}(t)\Big(\Sigma_i^{m,r_m}(t)\Big)^2 V_i^{m,r_m}(t)^\top = U_{i+1}^{m,r_m}(t)\Big(\Sigma_{i+1}^{m,r_m}(t)\Big)^2 U_{i+1}^{m,r_m}(t), \tag{10}
$$

and so we have two orthogonal eigenvalue decompositions of the same matrix. Since the singular values are positives and ordered in decreasing order, then $\Sigma_i^{m,r_m}(t) = \Sigma_{i+1}^{m,r_m}(t)$. Thus all the matrices $A_i^{m,r_m}(t)$, $\forall i$, have the same singular values. Let $\Sigma^{m,r_m}(t)$ be the diagonal matrix of singular values. It can be written as follows: $\Sigma^{m,r_m}(t) = \operatorname{diag}\big(\lambda_1^{m,r_m}(t)I_{\alpha_1},\dots,\lambda_p^{m,r_m}(t)I_{\alpha_p}\big)$, where, $\forall s \in \{1,\dots,p\}$, α_s is the multiplicity of the singular value $\lambda_s^{m,r_m}(t)$ and I_{α_s} is the $\alpha_s \times \alpha_s$ identity matrix. Moreover, (10) also implies that

$$
U_{i+1}^{m,r_m}(t) = V_i^{m,r_m}(t)O_i^{m,r_m}(t),
$$

where $O_i^{m,r_m}(t) = \operatorname{diag}\big(O_{i,1}^{m,r_m}(t),\dots,O_{i,p}^{m,r_m}(t)\big)$ and, $\forall s \in \{1,\dots,p\}$, $O_{i,s}^{m,r_m}(t) \in \mathbb{R}^{\alpha_s\times\alpha_s}$ is an orthogonal matrix. Using this and the fact that, $\forall i$, $O_i^{m,r_m}(t)$ and $\Sigma^{m,r_m}(t)$ commute, we obtain that

$$
A^{m,r_m}(t) := \prod_{i=1}^{k_m} A_i^{m,r_m}(t) = U_1^{m,r_m}(t)\prod_{i=1}^{k_m-1} O_i^{m,r_m}(t)\big(\Sigma^{m,r_m}(t)\big)^{k_m} V_{k_m}^{m,r_m}(t)^\top. \tag{11}
$$

This is a singular value decomposition of $A^{m,r_m}(t)$. So if $\Sigma^{m,r_m}(t) = \operatorname{diag}\big(\sigma_1^{m,r_m}(t),\dots,\sigma_{d_m}^{m,r_m}(t)\big)$, then the set of singular values of $A^{m,r_m}(t)$ is $\Big\{\big(\sigma_l^{m,r_m}(t)\big)^{k_m}, 1 \leq l \leq d_m\Big\}$. The matrix $A^{m,r_m}(t)$ can thus be written as follows

$$
A^{m,r_m}(t) = \sum_{l=1}^{d_m}\big(\sigma_l^{m,r_m}(t)\big)^{k_m} u_l^{m,r_m}(t)v_l^{m,r_m}(t)^\top,
$$

where $u_l^{m,r_m}(t)$ and $v_l^{m,r_m}(t)$ are the l-th left and right singular vectors of $A^{m,r_m}(t)$, respectively.

We are now able to compute a lower bound for $\left\|\prod\limits_{i=1}^{k_m} A_i^{m,r_m}(t) w_{r_m}^m(t)\right\|^2$.

$$\begin{aligned}
\left\|\prod_{i=1}^{k_m} A_i^{m,r_m}(t) w_{r_m}^m(t)\right\|^2 &= \left\|\sum_{l=1}^{d_m} \left(\sigma_l^{m,r_m}(t)\right)^{k_m} u_l^{m,r_m}(t) v_l^{m,r_m}(t)^\top w_{r_m}^m(t)\right\|^2 \\
&= \left\|\sum_{l=1}^{d_m} \left(\sigma_l^{m,r_m}(t)\right)^{k_m} \left\langle v_l^{m,r_m}(t), w_{r_m}^m(t)\right\rangle u_l^{m,r_m}(t)\right\|^2 \\
&= \sum_{l=1}^{d_m} \left(\sigma_l^{m,r_m}(t)\right)^{2k_m} \left\langle v_l^{m,r_m}(t), w_{r_m}^m(t)\right\rangle^2 \\
&\geq \left(\sigma_1^{m,r_m}(t)\right)^{2k_m} \left\langle v_1^{m,r_m}(t), w_{r_m}^m(t)\right\rangle^2 \\
&= \|A_i^{m,r_m}(t)\|_2^{2k_m} \left\langle v_1^{m,r_m}(t), w_{r_m}^m(t)\right\rangle^2 \\
&\geq \left(\frac{\|A_i^{m,r_m}(t)\|}{\sqrt{d_m}}\right)^{2k_m} \left\langle v_1^{m,r_m}(t), w_{r_m}^m(t)\right\rangle^2 .
\end{aligned}$$

Let us now recall that $\varepsilon_2(t) = \max\limits_{\substack{m=1\dots N\\ r_m=1\dots d_m\\ i=1\dots k_m}} \left|\|\omega_{r_m}^m(t)\|^2 - \|A_i^{m,r_m}(t)\|^2\right|$. If $\varepsilon(0) = 0$, then $\varepsilon_2(0) = 0$. Moreover, by Lemma 4.1 (ii), we have $\varepsilon_2(t)$ is constant over time. Then $\varepsilon_2(t) = 0,\ \forall t \geq 0$, which implies that, $\forall i \in \llbracket 1, k_m \rrbracket$, $\forall t \geq 0$, $\|A_i^{m,r_m}(t)\| = \|w_{r_m}^m(t)\|$. Using (8), we obtain that, $\forall i \in \llbracket 1, k_m \rrbracket$, $\forall t \geq 0$, $\|A_i^{m,r_m}(t)\| = \|\mathcal{G}_{:\dots:r_m:\dots:}(t)\|$. We then have

$$\left\|\prod_{i=1}^{k_m} A_i^{m,r_m}(t) w_{r_m}^m(t)\right\| \geq \left(\frac{\|\mathcal{G}_{:\dots:r_m:\dots:}(t)\|}{\sqrt{d_m}}\right)^{k_m} \left|\left\langle v_1^{m,r_m}(t), w_{r_m}^m(t)\right\rangle\right| . \tag{12}$$

On the other hand,

$$\left\|\prod_{i=1}^{k_m} A_i^{m,r_m}(t) w_{r_m}^m(t)\right\| \leq \left\|\prod_{i=1}^{k_m} A_i^{m,r_m}(t)\right\| \left\|w_{r_m}^m(t)\right\| \leq \prod_{i=1}^{k_m} \|A_i^{m,r_m}(t)\| \left\|w_{r_m}^m(t)\right\| = \|\mathcal{G}_{:\dots:r_m:\dots:}(t)\|^{k_m} \left\|w_{r_m}^m(t)\right\| . \tag{13}$$

i. The proof consists of applying Theorem 4.3 with (12) and (13) and using that $\delta_{r_m} \geq 0$ if $\dfrac{d}{dt}\|\mathcal{G}_{:\dots:r_m:\dots:}(t)\| \geq 0$.

ii. If $\dfrac{d}{dt}\|\mathcal{G}_{:\dots:r_m:\dots:}(t)\| \leq 0$, then $\delta_{r_m} \leq 0$, so the inequalities are reversed.

iii. Using the definition of $\delta_{r_m}(t)$ provided in Theorem 4.3, we have

$$\begin{aligned}
|\delta_{r_m}(t)| &= \left|\sum_{r_1,\dots,r_{m-1},r_{m+1},\dots,r_N} \mathcal{G}_{r_1,\dots,r_N}(t) \left(\prod_{n\neq m} \left\|\prod_{j=1}^{k_n} A_j^{n,r_n}(t)\omega_{r_n}^n(t)\right\|\right) \Delta_{r_1,\dots,r_N}(t)\right| \\
&\leq \sum_{r_1,\dots,r_{m-1},r_{m+1},\dots,r_N} |\mathcal{G}_{r_1,\dots,r_N}(t)| \left(\prod_{n\neq m} \left\|\prod_{j=1}^{k_n} A_j^{n,r_n}(t)\omega_{r_n}^n(t)\right\|\right) |\Delta_{r_1,\dots,r_N}(t)| \\
&\leq \sum_{r_1,\dots,r_{m-1},r_{m+1},\dots,r_N} |\mathcal{G}_{r_1,\dots,r_N}(t)| |\Delta_{r_1,\dots,r_N}(t)| \qquad \left(\text{since } \prod_{n\neq m} \left\|\prod_{j=1}^{k_n} A_j^{n,r_n}(t)\omega_{r_n}^n(t)\right\| \leq 1\right) \\
&\leq \sqrt{\sum_{r_1,\dots,r_{m-1},r_{m+1},\dots,r_N} \mathcal{G}^2_{r_1,\dots,r_N}(t)} \sqrt{\sum_{r_1,\dots,r_{m-1},r_{m+1},\dots,r_N} \Delta^2_{r_1,\dots,r_N}(t)} \qquad \text{(using Cauchy–Schwarz inequality)} \\
&\leq \|\mathcal{G}_{:\dots:r_m:\dots:}(t)\| \sup_{t\in[0,T]} \sqrt{\sum_{r_1,\dots,r_{m-1},r_{m+1},\dots,r_N} \Delta^2_{r_1,\dots,r_N}(t)}.
\end{aligned}$$

The last inequality holds because $\mathcal{L}$ is locally smooth, and so $t \mapsto \sqrt{\sum_{r_1,\ldots,r_{m-1},r_{m+1},\ldots,r_N} \Delta^2_{r_1,\ldots,r_N}(t)}$ is continuous and also bounded on the compact interval $[0,T]$. □

A.4 Proof of Corollary 4.5

Proof. The proof consists of applying Theorem 4.3 with $k_1 = \ldots = k_N = 0$ and using the fact that $\|\omega^m_{r_m}(t)\| = \|\mathcal{G}_{:\ldots:r_m:\ldots:}(t)\|$. □

A.5 Proof of Proposition 4.6

Proof. Let $\tilde{\mathcal{G}}$ the tensor obtained by setting to zero the $d_n - R_n$ subtensors of G in the mode n satisfying:

$$\|G_{:\ldots:r_n:\ldots:}\| \leq \frac{\epsilon}{\sqrt{d_n N}\prod_{n=1}^N \|V^{(n)}\|_2}, \quad \forall n \in [\![1, N]\!].$$

We assume without loss of generality that these subtensors are in positions $R_n + 1, \ldots, d_n$. Let $\tilde{\mathcal{W}}$ be the tensor computed as follows $\tilde{\mathcal{W}} = \tilde{\mathcal{G}} \times_1 V^{(1)} \times_2 V^{(2)} \cdots \times_N V^{(N)}$. For all $n \in [\![1, N]\!]$, $\tilde{\mathcal{G}}$ has $d_n - R_n$ subtensors with zero norm. The matricization $[\tilde{\mathcal{G}}]_{(n)}$, $\forall n \in [\![1, N]\!]$, has then $d_n - R_n$ rows with zero norm, and so $\mathrm{rank}([\tilde{\mathcal{G}}]_{(n)}) \leq R_n$, $\forall n \in [\![1, N]\!]$. On the other hand, applying mode-n matricization to $\tilde{\mathcal{W}}$, we obtain

$$[\tilde{\mathcal{W}}]_{(n)} = V^{(n)}[\tilde{\mathcal{G}}]_{(n)} \left(V^{(N)} \odot \ldots \odot V^{(n+1)} \odot V^{(n-1)} \odot \ldots V^{(1)}\right)^\top.$$

Thus, $\forall n \in [\![1, N]\!]$, $\mathrm{rank}([\tilde{\mathcal{W}}]_{(n)}) \leq \mathrm{rank}([\tilde{\mathcal{G}}]_{(n)}) \leq R_n$. The multilinear rank of $\tilde{\mathcal{W}}$ is then less than $(R_1, \ldots, R_n)$.

Moreover,

$$\begin{aligned} \|\mathcal{W}_e - \tilde{\mathcal{W}}\| &= \|(\mathcal{G} - \tilde{\mathcal{G}}) \times_1 V^{(1)} \times_2 V^{(2)} \cdots \times_N V^{(N)}\| \\ &\leq \|\mathcal{G} - \tilde{\mathcal{G}}\| \|V^{(1)}\|_2 \ldots \|V^{(N)}\|_2. \end{aligned} \tag{14}$$

We have:

$$\begin{aligned} \|\mathcal{G} - \tilde{\mathcal{G}}\|^2 &\leq \sum_{n=1}^N \sum_{r_n=R_n+1}^{d_n} \|G_{:\ldots:r_n:\ldots:}\|^2 \\ &\leq \sum_{n=1}^N \frac{d_n - R_n}{d_n N} \frac{\epsilon^2}{\|V^{(1)}\|_2^2 \ldots \|V^{(N)}\|_2^2} \\ &\leq \frac{\epsilon^2}{\|V^{(1)}\|_2^2 \cdots \|V^{(N)}\|_2^2}. \end{aligned}$$

Then, using (14), we obtain $\|\mathcal{W}_e - \tilde{\mathcal{W}}\| \leq \epsilon$. Thus, we found a tensor $\tilde{\mathcal{W}}$ such as $\|\mathcal{W}_e - \tilde{\mathcal{W}}\| \leq \epsilon$ and $\mathrm{multirank}(\tilde{\mathcal{W}}) \leq (\mathrm{R}_1, \ldots, \mathrm{R}_\mathrm{n})$. This completes the proof. □