Machine Learning (2021) 110:1867–1900
https://doi.org/10.1007/s10994-021-05987-8

Tensor Q-rank: new data dependent definition of tensor rank

Hao Kong[1] · **Canyi Lu**[2] · **Zhouchen Lin**[1]

Received: 17 September 2020 / Revised: 5 April 2021 / Accepted: 24 April 2021 /
Published online: 21 June 2021

Abstract
Recently, the *Tensor Nuclear Norm* (*TNN*) regularization based on t-SVD has been widely used in various low tubal-rank tensor recovery tasks. However, these models usually require smooth change of data along the third dimension to ensure their low rank structures. In this paper, we propose a new definition of data dependent tensor rank named *tensor Q-rank* by a learnable orthogonal matrix $\mathbf{Q}$, and further introduce a unified data dependent low rank tensor recovery model. According to the low rank hypothesis, we introduce two explainable selection methods of $\mathbf{Q}$, under which the data tensor may have a more significant low tensor Q-rank structure than that of low tubal-rank structure. Specifically, maximizing the variance of singular value distribution leads to Variance Maximization Tensor Q-Nuclear norm (VMTQN), while minimizing the value of nuclear norm through manifold optimization leads to Manifold Optimization Tensor Q-Nuclear norm (MOTQN). Moreover, we apply these two models to the low rank tensor completion problem, and then give an effective algorithm and briefly analyze why our method works better than TNN based methods in the case of complex data with low sampling rate. Finally, experimental results on real-world datasets demonstrate the superiority of our proposed models in the tensor completion problem with respect to other tensor rank regularization models.

Keywords Tensor rank · Low rank · Tensor completion · Convex optimization

Editor: Nick Vannieuwenhoven.

✉ Zhouchen Lin
zlin@pku.edu.cn

Hao Kong
konghao@pku.edu.cn

Canyi Lu
canyilu@gmail.com

[1] Key Lab. of Machine Perception (MOE), School of EECS, Peking University, Beijing, China

[2] Department of Electrical & Computer Engineering (ECE), Carnegie Mellon University, Pittsburgh, USA

1 Introduction

With the development of data science, multi-dimensional data structures are becoming more and more complex. The low-rank tensor recovery problem, which aims to recover a low-rank tensor from an observed tensor, has also been extensively studied and applied. The problem can be formulated as the following model:

$$\min_{\mathcal{X}} \ \text{rank}(\mathcal{X}), \quad s.t.\ \Psi(\mathcal{X}) = \mathcal{Y}, \tag{1}$$

where $\mathcal{Y}$ is the observed measurement by a linear operator $\Psi(\cdot)$ and $\mathcal{X}$ is the clean data. Generally, it is difficult to solve Eq. (1) directly, and different rank definitions correspond to different models. The commonly used definitions of tensor rank are all related to particular tensor decompositions (Kolda and Bader 2009). For example, CP-rank (Hitchcock 1927) is based on the CANDECOMP/PARAFAC decomposition (Kiers 2000); multilinear rank (Hitchcock 1928) is based on the Tucker decomposition (Tucker 1966); tensor multi-rank and tubal-rank (Kilmer et al. 2013) are based on t-SVD (Kilmer and Martin 2011); and a new tensor rank with invertible linear operator (Lu et al. 2019) is based on T-SVD (Kernfeld et al. 2015). Among them, CP-rank and multilinear rank are both older and more widely studied, while the remaining two mentioned here are relatively new. Minimizing the rank function in Eq. (1) directly is usually NP-hard and is difficult to be solved within polynomial time, hence we often replace rank($\mathcal{X}$) by a convex/non-convex surrogate function. Similar to the matrix case (Candès and Recht 2009; Candès and Tao 2010), with different definitions of tensor singular values, various tensor nuclear norms are proposed as the rank surrogates (Liu et al. 2013; Friedland and Lim 2018; Kilmer and Martin 2011; Lu et al. 2019).

1.1 Existing mainstream methods and their limitations

Friedland and Lim (2018) introduce cTNN (Tensor Nuclear Norm based on CP) as the convex relaxation of the tensor CP-rank:

$$\|\mathcal{T}\|_{cTNN} = \inf\left\{ \sum_{i=1}^{r} |\lambda_i| \ : \ \mathcal{T} = \sum_{i=1}^{r} \lambda_i \mathbf{u}_i \circ \mathbf{v}_i \circ \mathbf{w}_i \right\}, \tag{2}$$

where $\|\mathbf{u}_i\| = \|\mathbf{v}_i\| = \|\mathbf{w}_i\| = 1$ and $\circ$ represents the vector outer product.[1] However, for a given tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, minimizing the surrogate objection $\|\mathcal{T}\|_{cTNN}$ directly is difficult due to the fact that computing CP-rank is usually NP-complete (Håstad 1990; Hillar and Lim 2013) and computing cTNN is NP-hard in some sense (Friedland and Lim 2018), which also mean we cannot verify the consistency of cTNN's implicit decomposition with the ground-truth CP-decomposition. Meanwhile, it is hard to measure the cTNN's tightness relative to the CP-rank.[2] Although Yuan and Zhang (2016) give the sub-gradient of cTNN by leveraging its dual property, the high computational cost makes it difficult to implement.

[1] Please see Kolda and Bader (2009) or our supplementary materials for more details.

[2] For the matrix case, the nuclear norm is the conjugate of the conjugate function of the rank function in the unit ball. However, it is still unknown whether this property holds for cTNN and CP-rank.

To reduce the computation cost of computing the rank surrogate function, Liu et al. (2013) define a kind of tensor nuclear norm named SNN (Sum of Nuclear Norm) based on the Tucker decomposition (Tucker 1966):

$$\|\mathcal{T}\|_{SNN} = \sum_{i=1}^{d} \left\|\mathbf{T}_{(i)}\right\|_*, \tag{3}$$

where $\mathcal{T} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, $\mathbf{T}_{(i)} \in \mathbb{R}^{(n_1 \ldots n_{i-1} n_{i+1} \ldots n_d) \times n_i}$ denotes unfolding the tensor along the ith dimension, and $\|\cdot\|_*$ is the nuclear norm of a matrix, i.e., sum of singular values. The convenient calculation algorithm makes SNN widely used (Fu et al. 2016; Liu et al. 2015, 2013; Kasai and Mishra 2016; Li et al. 2016). It is worth to mentioned that, although SNN has a similar representation to matrix case, Romera-Paredes and Pontil (2013) point out that SNN is not the tightest convex relaxation of the multilinear rank (Hitchcock 1928), and is actually an overlap regularization of it. References Tomioka et al. (2010); Tomioka and Suzuki (2013); Wimalawarne et al. (2014) also propose a new regularizer named Latent Trace Norm to better approximate the tensor rank function. In addition, due to unfolding the tensor directly along each dimension, the information utilization of SNN based model is insufficient.

To avoid information loss in SNN, Kilmer and Martin (2011) propose a tensor decomposition named t-SVD with a Fourier transform matrix $\mathbf{F}$, and Zhang et al. (2014) give a definition of the tensor nuclear norm on $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ corresponding to t-SVD, i.e., Tensor Nuclear Norm (TNN):

$$\|\mathcal{T}\|_{TNN} := \frac{1}{n_3} \sum_{i=1}^{n_3} \left\|\mathbf{G}^{(i)}\right\|_*, \quad \text{where } \mathcal{G} = \mathcal{T} \times_3 \mathbf{F}, \tag{4}$$

where $\mathbf{G}^{(i)}$ denotes the ith frontal slice matrix of tensor $\mathcal{G}$,[3] and $\times_3$ is the mode-3 multilinear multiplication (Tucker 1966). Benefitting from the efficient Discrete Fourier transform and the better sampling effect of Fourier basis on time series features, TNN has attracted extensive attention in recent years (Zhang et al. 2014; Lu et al. 2016, 2018; Yin et al. 2018; Hu et al. 2016). The operation of Fourier transform along the third dimension makes TNN based models have a natural computing advantage for video and other data with strong time continuity along a certain dimension.

However, when considering the smoothness of different data, using a fixed Fourier transform matrix $\mathbf{F}$ may bring some limitations. In this paper, we define smooth and non-smooth data along a certain dimension as the usual intuitive meaning, which means the slices of tensor data along a dimension are arranged in a certain paradigm, e.g., time series. For example, a continuous video data is smooth. But if the data tensor is a concatenation of several different scene videos or a random arrangement of all frames, then the data is non-smooth.

Firstly, TNN needs to implement Singular Value Decomposition (SVD) in the complex field $\mathbb{C}$, which is slightly slower than that in the real field $\mathbb{R}$. Besides, the experiments in related papers (Zhang et al. 2014; Lu et al. 2018; Zhou et al. 2018; Kong et al. 2018) are usually based on some special dataset which have smooth change along the third dimension, such as RGB images and short videos. Those non-smooth data may increase the

[3] The implementation of Fourier transform along the third dimension of $\mathcal{T}$ is equivalent to multiplying a DFT matrix $\mathbf{F}$ by using $\times_3$. For more details, please see Sect. 2.2.

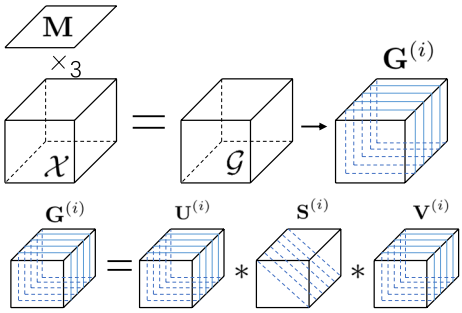

Fig. 1 Replace **F** in Eq. (4) by matrix **M** and further obtain new definitions of tensor rank $\text{rank}_M(\mathcal{X})$ and tensor nuclear norm $\|\mathcal{X}\|_{M,*}$ by using $\mathbf{S}^{(i)}$

number of non-zero tensor singular values (Kilmer and Martin 2011; Zhang et al. 2014), weakening the significance of low rank structure. Since tensor multi-rank (Zhang et al. 2014) is actually the rank of each projection matrix on different Fourier basis, the non-smooth change along the third dimension may lead to large singular values appearing on the projection matrix slices which are corresponding to the high frequency.

1.2 Related work

In order to solve the above phenomenon, there are some works (Kernfeld et al. 2015; Xu et al. 2019; Song et al. 2019; Lu et al. 2019; Jiang et al. 2020) that consider to improve the projection matrix of TNN, i.e., the Discrete Fourier transform matrix **F** in Eq. (4). These work want to replace **F** by another measurement matrix **M** and further obtain new definitions of tensor rank $\text{rank}_M(\mathcal{X})$ and tensor nuclear norm $\|\mathcal{X}\|_{M,*}$ as regularizers. Figure 1 shows the related operations. Their recovery models can be summarized as follows:

$$\min_{\mathcal{X}} \|\mathcal{X}\|_{M,*}, \quad s.t.\ \Psi(\mathcal{X}) = \mathcal{Y},\ \mathbf{M} \text{ is determined by some prior knowledge.} \tag{5}$$

Please see Sect. 2 for the relevant definitions in Eq. (5). In the following, we will discuss the motivations and limitations of these work (Kernfeld et al. 2015; Xu et al. 2019; Song et al. 2019; Lu et al. 2019; Jiang et al. 2020), respectively.

Kernfeld et al. (2015) generalize the t-product by introducing a new operator named cosine transform product with an **arbitrary invertible** linear transform $\mathcal{L}$ (or **arbitrary invertible** matrix **M**). For a given $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and an invertible matrix $\mathbf{M} \in \mathbb{R}^{n_3 \times n_3}$, they have $\mathcal{L}_{\mathbf{M}}(\mathcal{T}) = \mathcal{T} \times_3 \mathbf{M}$ and $\mathcal{L}^{-1}_{\mathbf{M}}(\mathcal{T}) = \mathcal{T} \times_3 \mathbf{M}^{-1}$. Different from the commonly used definition of tensor mode-i product in Kolda and Bader (2009); Liu et al. (2013); Kernfeld et al. (2015); Lu et al. (2019), it should be mentioned that for convenience in this paper, we define $\mathcal{L}_{\mathbf{Q}}(\mathcal{T}) = \mathcal{T} \times_3 \mathbf{Q} = \text{fold}_3(\mathbf{T}_{(3)}\mathbf{Q})$, where $\mathbf{T}_{(3)} \in \mathbb{R}^{n_1 n_2 \times n_3}$ and is defined by $\mathbf{T}_{(3)} := \text{unfold}_3(\mathcal{T})$. That is to say, we arrange the tensor fiber $\mathcal{T}_{ij:}$ by rows.

Following this idea, Lu et al. (2019) propose a new tensor nuclear norm induced by invertible linear transforms (Kernfeld et al. 2015). Different from Kilmer and Martin (2011); Zhang et al. (2014), they use an fixed invertible matrix to replace the Fourier transform matrix in TNN. Although this method improves the performance of the recovery model to a certain extent, some new problems still arise, such as how to determine the fixed invertible matrix. Normally, different data need different optimal invertible matrix, but a reasonable matrix selection method is not given in Lu et al. (2019). Furthermore, the Frobenius norm of the invertible matrix is uncertain, which may lead to some computational problems, e.g., approaching zero or infinity.

Additionally, Kernfeld et al. (2015) propose an idea that, with the help of Toeplitz-plus-Hankel matrix (Ng et al. 1999), the Discrete cosine transform matrix $\mathbf{C}$ can also be used to replace $\mathbf{F}$. Then the work Xu et al. (2019) propose some fast algorithms for diagonalization and the relevant recovery model. However, $\mathbf{C}$ is still based on trigonometric function, and may lead to the similar problems with TNN based model, as we mentioned in the last paragraph of Sect. 1.1.

Considering the efficiency of time-space transformation, the work Song et al. (2019) use the Daubechies 4 discrete wavelet transform matrix to replace $\mathbf{F}$. As we know, the wavelet transform can take the position information into account, which may make it better than Fourier transform and cosine transform in handling some special data, e.g., audio data. However, many wavelet bases generate transform matrices in exponential form, which means the large scale wavelet matrix may bring the problem of computational complexity.

Regardless of the computational complexity, Jiang et al. (2020) introduce a new projection matrix named tight framelets transform matrix (Cai et al. 2008; Jiang et al. 2018). They claim that redundancy in the transformation is important as such transformed coefficients can contain information of missing data in the original domain (Cai et al. 2008). However, we consider that redundancy is not a sufficient condition to improve the effect of recovery model shown in Eq. (5).

In summary, different multipliers $\mathbf{M}$ in Eq. (5) lead to different definitions of regularizer, which may lead to different experimental results. However, there is still no unified rules for selecting $\mathbf{M}$. It can be seen from the above methods that when $\mathbf{M}$ is selected as orthogonal matrix, it is convenient for calculation and interpretation. In general, projection bases are unit orthogonal. We further think that each kind of data should have its best matching matrix, i.e., $\mathbf{M}$ could be data dependent. In this paper, we solve the problem of how to define a better data dependent orthogonal transformation matrix.

1.3 Motivation

In the tensor completion task, we find that when dealing with some non-smooth data, Tensor Nuclear Norm (TNN) based methods usually perform worse than the cases with smooth data. Therefore, we want to improve this phenomenon by changing the projection basis $\mathbf{F}$ in Eq. (4). In other words, we provide some interpretable selection criteria of $\mathbf{M}$ in Eq. (5), e.g., make $\mathbf{M}$ be an orthogonal matrix and data dependent w.r.t. the data tensor $\mathcal{X}$. The following gives the details:

$$\min_{\mathcal{X},\mathbf{Q}} \|\mathcal{X}\|_{Q,*}, \quad s.t.\ \Psi(\mathcal{X}) = \mathcal{Y},\ \mathbf{Q}^\top\mathbf{Q} = \mathbf{I},\ \mathbf{Q} \text{ is determined by } \mathcal{X}. \tag{6}$$

Whether in the case of matrix recovery (Candès and Recht 2009; Candès and Tao 2010) or tensor recovery (Zhang and Aeron 2017; Lu et al. 2018, 2019), the low rank hypothesis is very important. Generally speaking, the lower the rank of the data, the easier it is to recover with fewer observations. As can be seen from Fig. 2, we can use a better $\mathbf{Q}$ to make the low rank structure of the non-smooth data more significant.

Considering the convex relaxation, the low rank property is usually represented by **(a): the distribution of singular values**, or **(b): the value of nuclear norm**. We may as well take these two points as priori knowledge respectively, and specify the selection rules of $\mathbf{Q}$ in Eq. (6), so that the low rank property of $\mathcal{X}$ can be better reflected. Therefore, we provide two methods in this paper as follows:

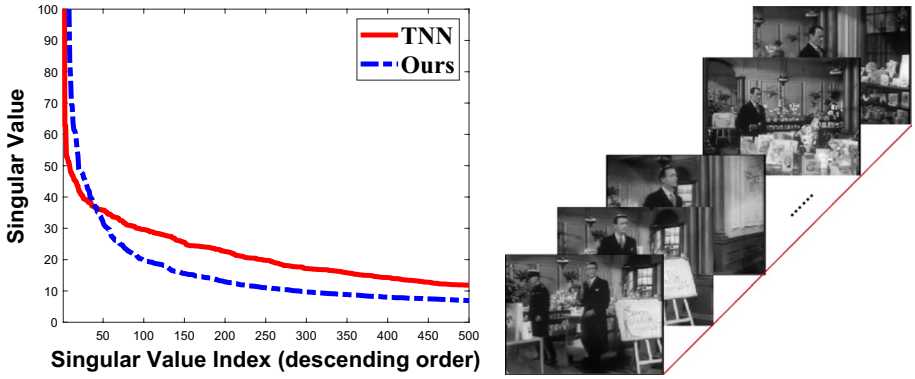

Fig. 2 Compare the two different low rank structures between our proposed regularization and TNN regularization in non-smooth video data. Left: the first 500 sorted singular values by TNN regularization (divided by $\sqrt{n_3}$) and ours. Right: the short video with background changes

(a): Let $\mathbf{Q}$ satisfy a certain selection method to make more tensor singular values close to 0 while the remaining ones are far from 0. From another perspective, the distribution variance of singular values should be larger, which leads to Variance Maximization Tensor Q-Nuclear norm (VMTQN) in Sect. 3.1.

(b): Let $\mathbf{Q}$ minimize the nuclear norm $\|\mathcal{X}\|_{Q,*}$ directly, leading to a bilevel problem. As we know, nuclear norm is usually used as an surrogate function of the rank function. Then we use some manifold optimization method to solve the problem, which leads to Manifold Optimization Tensor Q-Nuclear norm (MOTQN) in Sect. 3.2.

1.4 Contributions

In summary, our main contributions include:

- We propose a unified data dependent low rank tensor recovery model which is shown in Eq. (6). Among them, the corresponding definitions of tensor Q-rank $\text{rank}_Q(\mathcal{X})$ and tensor Q-nuclear norm $\|\mathcal{X}\|_{Q,*}$ are proposed along with the learnable data dependent orthogonal $\mathbf{Q}$.
- From the low rank hypothesis, we consider the distribution of singular values and the value of nuclear norm as prior knowledge respectively, leading to two different selection rules of $\mathbf{Q}$. It should be noted that both methods are designed to make the low rank structure more significant. Figure 2 shows an example with background changing video data that, under our proposed selection of $\mathbf{Q}$, our low rank structure is more significant.
- For each method, we give relatively complete theoretical derivations, including interpretation and optimization. As for VMTQN in Sect. 3.1, we start from variance maximization and use Theorem 2 to associate $\ell_{2,1}$ norm minimization with singular value decomposition, and further make $\mathbf{Q}$ select as the matrix of right singular vectors. On the other hand, MOTQN in Sect. 3.2 minimizes the nuclear norm directly and use manifold optimization algorithm to update $\mathbf{Q}$ in each iteration.
- Finally, we apply our proposed regularizers with adaptive $\mathbf{Q}$ to the tensor completion problem. We analyze the computational complexity, convergence and performance guarantee of our algorithm to a certain extent. Moreover, we explain why the more sig-

nificant the low rank structure, the easier the data can be recovered, which corresponds to our motivation.

2 Notations and preliminaries

2.1 Notations

We introduce some notations and necessary definitions which will be used later. Tensors are represented by uppercase calligraphic letters, e.g., $\mathcal{T}$. Matrices are represented by boldface uppercase letters, e.g., $\mathbf{M}$. Vectors are represented by boldface lowercase letters, e.g., $\mathbf{v}$. Scalars are represented by lowercase letters, e.g., s. Given a third-order tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, we use $\mathbf{T}^{(k)}$ to represent its kth frontal slice $\mathcal{T}(:,:,k)$ while its (i,j,k)th entry is represented as $\mathcal{T}_{ijk}$. $\sigma_i(\mathbf{X})$ denotes the ith largest singular value of matrix $\mathbf{X}$. $\mathbf{X}^+$ denotes the pseudo-inverse matrix of $\mathbf{X}$. $\|\mathbf{X}\|_\sigma = \sigma_1(\mathbf{X})$ denotes the matrix spectral norm. $\|\mathbf{X}\|_* = \sum_{i=1}^{\min\{n_1,n_2\}} \sigma_i(\mathbf{X})$ denotes the matrix nuclear norm and $\|\mathbf{X}\|_{2,1} = \sum_{j=1}^{n_2} \sqrt{\sum_{i=1}^{n_1} \mathbf{X}_{ij}^2}$ denotes the matrix $\ell_{2,1}$ norm, where $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2}$ and $\mathbf{X}_{ij}$ is the (i,j)th entry of $\mathbf{X}$.

$\mathbf{T}_{(3)} \in \mathbb{R}^{n_1 n_2 \times n_3}$ denotes unfolding the tensor $\mathcal{T}$ along the 3rd dimension by rows, which is little different from Kolda and Bader (2009); Kernfeld et al. (2015). That is to say, we arrange the tensor fiber $\mathcal{T}_{ij:}$ by rows. We then define $\mathcal{L}_{\mathbf{Q}}(\mathcal{T}) = \mathcal{T} \times_3 \mathbf{Q} = \text{fold}_3(\mathbf{T}_{(3)}\mathbf{Q})$ and have $\mathcal{L}_{\mathbf{Q}}^{-1}(\mathcal{T}) = \mathcal{T} \times_3 \mathbf{Q}^{-1}$, where $\mathbf{T}_{(3)} \in \mathbb{R}^{n_1 n_2 \times n_3}$ and is defined by $\mathbf{T}_{(3)} := \text{unfold}_3(\mathcal{T})$. Due to limited space, for the definitions of $\mathcal{P}_{\mathcal{T}}$ (Lu et al. 2016), multilinear multiplication (Tucker 1966), t-product (Kilmer and Martin 2011), and so on, please see our Supplementary Materials.

2.2 Tensor Q-rank

For a given tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and a Fourier transform matrix $\mathbf{F} \in \mathbb{C}^{n_3 \times n_3}$, if we use $\mathbf{G}^{(i)}$ to represent the ith frontal slice of tensor $\mathcal{G}$, then the tensor multi-rank and Tensor Nuclear Norm (TNN) of $\mathcal{X}$ can be formulated by mode-3 multilinear multiplication as follows:

$$\text{rank}_m := \left\{ (r_1, \ldots, r_{n_3}) \;\middle|\; r_i = \text{rank}(\mathbf{G}^{(i)}), \mathcal{G} = \mathcal{X} \times_3 \mathbf{F} \right\}, \tag{7}$$

$$\|\mathcal{X}\|_* := \frac{1}{n_3} \sum_{i=1}^{n_3} \left\| \mathbf{G}^{(i)} \right\|_*, \quad \text{where } \mathcal{G} = \mathcal{X} \times_3 \mathbf{F}. \tag{8}$$

Comparing with CP-rank and cTNN mentioned in Sect. 1.1, it is quite easy to calculate Eqs. (7) and (8) through the matrix Singular Value Decomposition (SVD). Kernfeld et al. (2015) generalize the t-product by introducing a new operator named cosine transform product with an arbitrary invertible linear transform $\mathcal{L}$ (or arbitrary invertible matrix $\mathbf{Q}$). For an invertible matrix $\mathbf{Q} \in \mathbb{R}^{n_3 \times n_3}$, they have $\mathcal{L}_{\mathbf{Q}}(\mathcal{X}) = \mathcal{X} \times_3 \mathbf{Q}$ and $\mathcal{L}_{\mathbf{Q}}^{-1}(\mathcal{X}) = \mathcal{X} \times_3 \mathbf{Q}^{-1}$.

Here, we further define the invertible multiplier $\mathbf{Q}$ as any general real orthogonal matrix. It is worth mentioning that the orthogonal matrix $\mathbf{Q}$ has two good properties: one is invertibility, the other is to keep Frobenius norm invariant, i.e., $\|\mathcal{X}\|_F = \|\mathcal{L}_{\mathbf{Q}}(\mathcal{X})\|_F$. Then we introduce a new definition of tensor rank named Tensor Q-rank.

Definition 1 (*Tensor Q-rank*) Given a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and a fixed real orthogonal matrix $\mathbf{Q} \in \mathbb{R}^{n_3 \times n_3}$, the tensor Q-rank of $\mathcal{T}$ is defined as the following:

$$\text{rank}_Q(\mathcal{X}) := \sum_{i=1}^{n_3} \text{rank}(\mathbf{G}^{(i)}), \quad \text{where } \mathcal{G} = \mathcal{L}_{\mathbf{Q}}(\mathcal{X}) = \mathcal{T} \times_3 \mathbf{Q}. \tag{9}$$

The corresponding low rank tensor recovery model can be written as follows:

$$\min_{\mathcal{X}} \text{rank}_Q(\mathcal{X}), \quad s.t.\ \Psi(\mathcal{X}) = \mathcal{Y}. \tag{10}$$

Generally in the low rank recovery models, due to the discontinuity and non-convexity of the rank function, it is quite difficult to minimize the rank function directly. Therefore, some auxiliary definitions of tensor singular value and tensor norm are needed to relax the rank function.

2.3 Definitions of tensor singular value and tensor norm

Considering the superior recovery performance of TNN in many existing tasks, e.g., video denoising (Lu et al. 2019) and subspace clustering (Yin et al. 2018), we can use the similar singular value definition of TNN. Given a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and a fixed orthogonal matrix $\mathbf{Q}$ such that $\mathcal{G} = \mathcal{L}_{\mathbf{Q}}(\mathcal{X})$, then the $\mathbf{Q}$-singular value of $\mathcal{X}$ is defined as $\{\sigma_j(\mathbf{G}^{(i)})\}$, where $i = 1, \ldots, n_3$, $j = 1, \ldots, \min\{n_1, n_2\}$, $\mathbf{G}^{(i)}$ is the i-the frontal slice of $\mathcal{G}$, and $\sigma(\cdot)$ denotes the matrix singular value. When an orthogonal matrix $\mathbf{Q}$ is fixed, the corresponding tensor spectral norm and tensor nuclear norm of $\mathcal{X}$ can also be given.

Definition 2 (*Tensor Q-spectral norm and Tensor Q-nuclear norm*) Given a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and a fixed real orthogonal matrix $\mathbf{Q} \in \mathbb{R}^{n_3 \times n_3}$, the tensor Q-spectral norm and tensor Q-nuclear norm of $\mathcal{X}$ are defined as the followings:

$$\|\mathcal{X}\|_{Q,\sigma} := \max_i \left\{ \left\| \mathbf{G}^{(i)} \right\|_\sigma \Big| \ \mathcal{G} = \mathcal{L}_{\mathbf{Q}}(\mathcal{X}) \right\}. \tag{11}$$

$$\|\mathcal{X}\|_{Q,*} := \sum_{i=1}^{n_3} \left\| \mathbf{G}^{(i)} \right\|_*, \quad \text{where } \mathcal{G} = \mathcal{L}_{\mathbf{Q}}(\mathcal{X}). \tag{12}$$

Moreover, with any fixed orthogonal matrix $\mathbf{Q}$, the convexity, duality, and envelope properties are all preserved.

Property 1 (Convexity) *Tensor Q-nuclear norm and Tensor Q-spectral norm are both convex.*

Property 2 (Duality) *Tensor Q-nuclear norm is the dual norm of Tensor Q-spectral norm, and vice versa.*

Property 3 (Convex envelope) *Tensor Q-nuclear norm is the tightest convex envelope of the Tensor Q-rank within the unit ball of the Tensor Q-spectral norm.*

These three properties are quite important in the low rank recovery theory. Property 3 implies that we can use the tensor Q-nuclear norm as a rank surrogate. That is to say, when

the orthogonal matrix $\mathbf{Q}$ is given, we can replace the low tensor Q-rank model (10) with model (13) to recover the original tensor:

$$\min_{\mathcal{X}} \ \|\mathcal{X}\|_{Q,*}, \quad s.t.\ \Psi(\mathcal{X}) = \mathcal{Y}. \tag{13}$$

In some cases, we will encounter the case that $\mathbf{Q}$ is not a square matrix, i.e., $\mathbf{Q} \in \mathbb{R}^{n_3 \times r}$ is column orthonormal. Then the corresponding definitions of $\text{rank}_Q(\mathcal{X})$ in Eq. (9) and $\|\mathcal{X}\|_{Q,*}$ in Eq. (12) also change to the sum of r frontal slices instead of n_3. Moreover, as for the convex envelope property, the double conjugate function of rank function $\text{rank}_Q(\mathcal{X})$ is still the corresponding nuclear norm $\|\mathcal{X}\|_{Q,*}$ within an unit ball. We give the following theorem to illustrate this case:

Theorem 1 *Given a tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and a fixed real column orthonormal matrix $\mathbf{Q} \in \mathbb{R}^{n_3 \times r}$. Let $\mathbf{Q}_\perp \in \mathbb{R}^{n_3 \times (n_3 - r)}$ be the column complement matrix of $\mathbf{Q}$,* and $\mathbf{Q}_t = \begin{bmatrix}\mathbf{Q} & \mathbf{Q}_\perp\end{bmatrix}$ *be a orthogonal matrix. Then within the unit ball $\mathcal{D} = \{\mathcal{X} | \|\mathcal{X}\|_{Q_t} \leq 1\}$, the double conjugate function of* $\text{rank}_Q(\mathcal{X})$ is $\|\mathcal{X}\|_{Q,*}$:

$$rank_Q^{**}(\mathcal{X}) = \|\mathcal{X}\|_{Q,*}. \tag{14}$$

In other words, $\|\mathcal{X}\|_{Q,*}$ is still the tightest convex envelope of $rank_Q^{**}(\mathcal{X})$ within the unit ball $\mathcal{D}$.

Theorem 1 indicate that even if $\mathbf{Q}$ is not a square matrix, Eq. (13) can still be used as an effective recovery model.

3 Two ways for determining Q: maximizing variance & Stiefel manifold optimization

In practical problems, the selection of $\mathbf{Q}$ often has a tremendous impact on the performance of the model (13). If $\mathbf{Q}$ is an identity matrix $\mathbf{I}$, it is equivalent to solving each frontal slice separately by the low rank matrix methods (Candès and Recht 2009). Or if $\mathbf{Q}$ is a Fourier transform matrix $\mathbf{F}$, it is equivalent to the TNN-based methods (Zhang et al. 2014; Lu et al. 2016; Zhang and Aeron 2017). Through the analysis of Lu et al. (2019) and our previous section, for a given data $\mathcal{X}$, those $\mathbf{Q}$ that make $\text{rank}_Q(\mathcal{X})$ lower usually make the recovery problem (13) easier.

Following, if we let $\mathbf{Q}$ in Eqs. (10) and (13) be a learnable variable w.r.t. data tensor $\mathcal{X}$, we can get a **data-dependent** tensor rank and corresponding low rank recovery model:

$$\min_{\mathcal{X},\mathbf{Q}} \ \|\mathcal{X}\|_{Q,*}, \quad s.t.\ \Psi(\mathcal{X}) = \mathcal{Y},\ \mathbf{Q} \text{ is determined by } \mathcal{X}. \tag{15}$$

Easy to see that Eq. (15) is actually a bilevel model and is usually hard to be solved directly. In the following, we will show two ways to solve this problem from the following two perspectives:

1. One is to use the prior knowledge of $\mathcal{X}$ to specify the selection criteria of $\mathbf{Q}$. For the low rank hypothesis, we usually measure it by the distribution of singular values. Therefore, we consider artificially specifying the conditions that $\mathbf{Q}$ should satisfy so as to maximize the variance of the corresponding singular values.

2. The other is to give the function $\mathbf{Q} = \text{argmin} f(\mathcal{X}, \mathbf{Q}) = \text{argmin} \|\mathcal{X}\|_{Q,*}$ and then use manifold optimization to solve the bilevel model directly. That is to say, We directly minimize the surrogate function of rank function (Property 3 and Theorem 1). It should be noted that although this way has higher rationality, it corresponds to a higher computational complexity.

From the above two perspectives, $\mathbf{Q}$ will be data dependent. In the following, we will introduce our two methods in two sub-sections respectively (Sects. 3.1 and 3.2). And in the last part (Sect. 3.3), considering a third-order tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, we analyze the applicability of each method in two different situations, i.e., $n_1 n_2 < n_3$ and $n_1 n_2 > n_3$.

3.1 Way I (VMTQN): specify the selection of Q by variance maximization

Let $\mathcal{G} = \mathcal{L}_{\mathbf{Q}}(\mathcal{X}) = \mathcal{X} \times_3 \mathbf{Q}$ and $\{\mathbf{G}^{(i)}\}_i$ denotes the frontal slices of $\mathcal{G}$. We hope to find a data-dependent $\mathcal{L}_{\mathbf{Q}}$ in Eqs. (12) and (13) instead of $\mathcal{L}_{\mathbf{F}}$ in TNN (Eq. (8)), which can reduce the number of non-zero singular values of each projected slices $\mathbf{G}^{(i)}$. Our analyses are as follows.

(1) If we make $\mathbf{Q}$ an orthogonal matrix, then it is also invertible. By using the unitary invariance of the Frobenius norm, the sum of the squares of each projected slice's Frobenius norm is a **constant** C, i.e., $\sum_{i=1}^{n_3} \|\mathbf{G}^{(i)}\|_F^2 = \|\mathcal{X}\|_F^2 = C$. Therefore, we need to consider how to select $\mathbf{Q}$ to make more singular values of $\{\mathbf{G}^{(i)}\}$ close to zero while the square sum of all singular values is a constant, i.e., $\sum_{j=1}^{n_3} \sigma_j(\mathbf{G}^{(i)})^2 = C$.

(2) Considering the definitions of tensor rank, tensor norm and tensor singular value corresponding to TNN in Zhang et al. (2014); Zhang and Aeron (2017), and tensor Q-rank in this paper, the matrix inequality $\frac{1}{n} \sum_{j=1}^{n} \sigma_j(\mathbf{G}^{(i)}) \leq \|\mathbf{G}^{(i)}\|_\sigma \leq \|\mathbf{G}^{(i)}\|_F$ (singular value, spectral norm and Frobenius norm, respectively) implies that, the closer $\|\cdot\|_F$ is to zero, the more singular values are close to zero, which will lead to a more significant tensor low rank structure (w.r.t. $\text{rank}_Q(\mathcal{X})$) with high probability.

3.1.1 From variance maximization to singular matrix

Combined with above two points, it is easy to see that we need to make more $\|\mathbf{G}^{(i)}\|_F$ close to 0 while the sum of squares $\sum_{i=1}^{n_3} \|\mathbf{G}^{(i)}\|_F^2$ is a constant C. From the perspective of variable distribution, we need to choose a data-dependent $\mathbf{Q}$ to maximize the distribution **variance** of $\{\|\mathbf{G}^{(i)}\|_F\}$, where $\mathcal{G} = \mathcal{L}_{\mathbf{Q}}(\mathcal{X})$ and $\mathbf{G}^{(i)}$ is the ith frontal slice matrix of $\mathcal{G}$. For better explanations, we give the following two lemmas, and the optimality condition of Lemma 1 illustrate our hypothesis that there should be more $\|\mathbf{G}^{(i)}\|_F$ close to 0.[4]

Lemma 1 *Given n non-negative variables $\{a_1, a_2, \ldots, a_n\}$ such that $\sum_{i=1}^{n} a_i^2 = C$, then maximizing the variance* $\text{Var}[a_i]$ *is equivalent to minimizing the summation $\sum_{i=1}^{n} a_i$. Moreover, the optimal condition is that there is only one non-zero variable in $\{a_1, a_2, \ldots, a_n\}$. Please see "Appendix A" for proof.*

[4] Notice that minimizing $\sum_{i=1}^{n} a_i$ in Lemma 1 can be seen as a linear hyperplane optimization problem defined in the first quadrant Euclidean spherical surface: $\{(a_1, \ldots, a_n) | \sum_{i=1}^{n} a_i^2 = C, a_i \geq 0\}$. The intersection of sphere and each axis is distributed on the optimal hyperplane, which corresponds to only one non-zero coordinate (more variables close to 0).

By using Lemma 1, maximizing the variance of $\{\|\mathbf{G}^{(i)}\|_F\}$ is equivalent to minimizing the sum $\sum_{i=1}^{n_3}\|\mathbf{G}^{(i)}\|_F$. Then we have $\sum_{i=1}^{n_3}\|\mathbf{G}^{(i)}\|_F = \|\mathbf{G}_{(3)}\|_{2,1} = \|\mathbf{X}_{(3)}\mathbf{Q}\|_{2,1}$, where $\mathbf{G}_{(3)}$ and $\mathbf{X}_{(3)}$ denote the mode-3 unfolding matrices (Tucker 1966).

Lemma 2 *Given a fixed matrix* $\mathbf{X} \in \mathbb{R}^{n_1\times n_2}$, *and its full Singular Value Decomposition as* $\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^\top$ *with* $\mathbf{U} \in \mathbb{R}^{n_1\times n_1}$, $\mathbf{\Sigma} \in \mathbb{R}^{n_1\times n_2}$, *and* $\mathbf{V} \in \mathbb{R}^{n_2\times n_2}$. *Then the matrix of right singular vectors* $\mathbf{V}$ *optimizes the following*:

$$\min_{\mathbf{Q}\in\mathbb{R}^{n_2\times n_2}} \|\mathbf{XQ}\|_{2,1}, \quad \text{s.t. } \mathbf{Q}^\top\mathbf{Q} = \mathbf{I}, \tag{16}$$

where $\|\mathbf{M}\|_{2,1} = \sum_{i=1}^{col}\|\mathbf{M}_{:,i}\|_2$ *is the sum of the* ℓ_2 *norms of all column vectors. Please see* "*Appendix* B" *for proof.*

Lemma 1 turns the maximizing variance problem into minimizing summation problem, while Lemma 2 gives a feasible solution to the problem of minimizing the summation of ℓ_2 norm. However, when $n_1 \le n_2$, there will be some zero-columns appearing in $\mathbf{\Sigma}$. We can use skinny SVD to reduce the redundant columns of $\mathbf{Q}$ in Eq. (16). Note that the size of $\mathbf{V}$ in skinny SVD is related to the size of $\mathbf{X}$. Considering the two cases $n_1 \ge n_2$ and $n_1 < n_2$ of $\mathbf{X} \in \mathbb{R}^{n_1\times n_2}$, we introduce an auxiliary variable $r = \min\{n_1, n_2\}$ to unify the matrix of right singular vectors as $\mathbf{V} \in \mathbb{R}^{n_2\times r}$. Furthermore, we need add an extra constraint $\mathbf{XQQ}^\top = \mathbf{X}$ to avoid the trivial solution when $r < n_2$. If not, $\mathbf{Q}$ may converge to the singular spaces which are corresponding to smaller singular values. For example, when $r = n_1 < n_2$ and $\mathbf{Q} \in \mathbb{R}^{n_2\times(n_2-r)}$, the optimal solution set of $\mathbf{Q}^*$ for Eq. (16) includes the null singular spaces of $\mathbf{X}$, which makes $\mathbf{XQ} = \mathbf{O}$ hold and the objective function value is 0. Then we have the following:

Theorem 2 *Given a fixed matrix* $\mathbf{X} \in \mathbb{R}^{n_1\times n_2}$ *with* $r = \min\{n_1, n_2\}$, *and its skinny Singular Value Decomposition as* $\mathbf{X} = \mathbf{U\Sigma V}^\top$ *where* $\mathbf{U} \in \mathbb{R}^{n_1\times r}$, $\mathbf{\Sigma} \in \mathbb{R}^{r\times r}$, *and* $\mathbf{V} \in \mathbb{R}^{n_2\times r}$. *Then the matrix of right singular vectors* $\mathbf{V}$ *optimizes the following*:

$$\min_{\mathbf{Q}\in\mathbb{R}^{n_2\times r}} \|\mathbf{XQ}\|_{2,1}, \quad \text{s.t. } \mathbf{Q}^\top\mathbf{Q} = \mathbf{I},\ \mathbf{XQQ}^\top = \mathbf{X}. \tag{17}$$

The proofs of the above please see "Appendix C". Theorem 2 shows that, to minimize the $\ell_{2,1}$ norm $\|\mathbf{X}_{(3)}\mathbf{Q}\|_{2,1}$ w.r.t. $\mathbf{Q}$, we can choose $\mathbf{Q}$ as the matrix of right singular vectors of $\mathbf{X}_{(3)}$.

3.1.2 Details of how to make Q data dependent

Through the analyses in Sect. 3.1.1, we make the selection of $\mathbf{Q}$ data-dependent, and the following definitions shows the details.

Definition 3 (*VMTQN: variance maximization tensor Q-nuclear norm*) Let $\mathcal{X} \in \mathbb{R}^{n_1\times n_2\times n_3}$ be a third-order tensor and $\mathbf{Q}$ be an orthogonal matrix. If $\mathcal{G} = \mathcal{X}\times_3 \mathbf{Q}$ and $\mathbf{G}^{(i)}$ denotes the frontal slices of $\mathcal{G}$, then the Variance Maximization Tensor Q-Nuclear norm (VMTQN) is defined as follows:

$$\|\mathcal{X}\|_{Q,*}, \text{ where } \mathbf{Q} = \underset{\mathbf{Q}^\top\mathbf{Q}=\mathbf{I}}{\operatorname{argmax}} \text{ Variance}\left\{\left\|\mathbf{G}^{(i)}\right\|_F\right\}. \tag{18}$$

Note that $\mathbf{Q}$ is determined by $\mathcal{X}$. With the help of Lemmas 1, 2, and Theorem 2, we can incorporate VMTQN into the low rank recovery model.

Definition 4 (*Low tensor Q-rank model with adaptive Q*) By setting the adaptive $\mathbf{Q}$ module as a low-level sub-problem, the low tensor Q-rank model (10) is transformed into the following:

$$\min_{\mathcal{X},\mathbf{Q}} \quad \text{rank}_Q(\mathcal{X}),\ s.t.\ \Psi(\mathcal{X}) = \mathcal{Y},\ \mathbf{Q} \in \underset{\mathbf{Q}^\top\mathbf{Q}=\mathbf{I}}{\text{argmin}} \|\mathbf{X}_{(3)}\mathbf{Q}\|_{2,1},\ \mathbf{X}\mathbf{Q}\mathbf{Q}^\top = \mathbf{X}. \tag{19}$$

And the corresponding surrogate model (13) is also replaced by the following:

$$\min_{\mathcal{X},\mathbf{Q}} \quad \|\mathcal{X}\|_{Q,*},\ s.t.\ \Psi(\mathcal{X}) = \mathcal{Y},\ \mathbf{Q} \in \underset{\mathbf{Q}^\top\mathbf{Q}=\mathbf{I}}{\text{argmin}} \|\mathbf{X}_{(3)}\mathbf{Q}\|_{2,1},\ \mathbf{X}\mathbf{Q}\mathbf{Q}^\top = \mathbf{X}. \tag{20}$$

In Eqs. (19) and (20), $\mathbf{X}_{(3)} \in \mathbb{R}^{n_1n_2\times n_3}$ denotes the mode-3 unfolding matrix of tensor $\mathcal{X} \in \mathbb{R}^{n_1\times n_2\times n_3}$, and $\mathbf{Q} \in \mathbb{R}^{n_3\times r}$ with $r = \min\{n_1n_2, n_3\}$.

Definition 5 In fact, Theorem 2 implies $\mathbf{Q} = \mathbf{V}$, where $\mathbf{V}$ is the matrix of right singular vectors of $\mathbf{X}_{(3)}$. If we let $\text{PCA}(\mathcal{X}, 3, r) := \underset{\mathbf{Q}^\top\mathbf{Q}=\mathbf{I}_r}{\text{argmin}} \|\mathbf{X}_{(3)}\mathbf{Q}\|_{2,1}$ be the operator to obtain the matrix of right singular vectors $\mathbf{Q} \in \mathbb{R}^{n_3\times r}$, where $r = \min\{n_1n_2, n_3\}$, then the models (19) and (20) can be abbreviated as follows:

$$\min_{\mathcal{X}} \ \text{rank}_Q(\mathcal{X}), \quad s.t.\ \Psi(\mathcal{X}) = \mathcal{Y},\ \mathbf{Q} = \text{PCA}(\mathcal{X}, 3, r). \tag{21}$$

$$\min_{\mathcal{X}} \ \|\mathcal{X}\|_{Q,*}, \quad s.t.\ \Psi(\mathcal{X}) = \mathcal{Y},\ \mathbf{Q} = \text{PCA}(\mathcal{X}, 3, r). \tag{22}$$

Remark 1 Notice that $\mathbf{Q} \in \mathbb{R}^{n_3\times r}$ in Eqs. (19) and (20) may not have full columns, i.e., $r < n_3$. The corresponding definitions of $\text{rank}_Q(\mathcal{X})$ in Eq. (9) and $\|\mathcal{X}\|_{Q,*}$ in Eq. (12) also change to the sum of r frontal slices instead of n_3. Then Theorem 1 guarantee the validity of Eq (20).

Remark 2 In fact, from "Appendix C" we can see that, r can be chosen as any value that satisfies the condition $\text{rank}(\mathbf{X}_{(3)}) \le r \le \min\{n_1n_2, n_3\}$, as long as $\mathbf{Q} \in \mathbb{R}^{n_3\times r}$ contains the whole column space of the matrix of right singular vectors $\mathbf{V}$ and is pseudo-invertible to make $\mathcal{X} = \mathcal{X}\times_3 \mathbf{Q}\times_3 \mathbf{Q}^+$ hold.

Within this framework, the orthogonal matrix $\mathbf{Q}$ is related to tensor $\mathcal{X}$. As we analyzed, choosing $\mathbf{Q}$ as the matrix of right singular vectors may make $\text{rank}_Q(\mathcal{X})$ as low as possible. In other words, there should be more "small" frontal slices of $\mathcal{X}\times_3 \mathbf{Q}$, whose Frobenius norms are close to 0 to guarantee the low tensor Q-rank structure of data with high probability.

Now the question is whether the function $\|\mathcal{X}\|_{Q,*}$ in Eq. (22) is still an envelope of the rank function $\text{rank}_Q(\mathcal{X})$ in Eq. (21) within an appropriate region. The following theorem shows that even if $\|\mathcal{X}\|_{Q,*}$ is no longer a convex function in the bilevel framework (22) since $\mathbf{Q}$ is dependent on $\mathcal{X}$, we can still use it as a surrogate for a lower bound of $\text{rank}_Q(\mathcal{X})$ in Eq. (21).

Theorem 3 *Given a column orthonormal matrix* $\mathbf{Q} \in \mathbb{R}^{n_3\times r}$, $r = \min\{n_1n_2, n_3\}$, *we use* $\text{rank}_{PCA}(\mathcal{X})$, $\|\mathcal{X}\|_{PCA,\sigma}$, *and* $\|\mathcal{X}\|_{PCA,*}$ to *abbreviate the corresponding concepts as follows*:

$$\text{rank}_{PCA}(\mathcal{X}) := \text{rank}_{Q}(\mathcal{X}), \text{where } \mathbf{Q} = \text{PCA}(\mathcal{X}, 3, r), \tag{23}$$

$$\|\mathcal{X}\|_{PCA,\sigma} := \|\mathcal{X}\|_{Q,\sigma}, \quad \text{where } \mathbf{Q} = \text{PCA}(\mathcal{X}, 3, r), \tag{24}$$

$$\|\mathcal{X}\|_{PCA,*} := \|\mathcal{X}\|_{Q,*}, \quad \text{where } \mathbf{Q} = \text{PCA}(\mathcal{X}, 3, r). \tag{25}$$

Then within the region of $\mathcal{D} = \{\mathcal{X} \mid \|\mathcal{X}\|_{PCA,\sigma} \leq 1\}$, *the inequality* $\|\mathcal{X}\|_{PCA,*} \leq \text{rank}_{PCA}(\mathcal{X})$ *holds. Moreover, for every fixed* $\mathbf{Q}$, *let* $\mathcal{S}_Q$ *denote the space* $\{\mathcal{X} \mid \mathbf{Q} \in \text{PCA}(\mathcal{X}, 3, r)\}$. *Then Theorem 1 indicates that* $\|\mathcal{X}\|_{PCA,*}$ *is still the tightest convex envelope of* $\text{rank}_{PCA}(\mathcal{X})$ *in* $\mathcal{S}_Q \cap \mathcal{D}$.

Remark 3 For any column orthonormal matrix $\mathbf{Q} \in \mathbb{R}^{n_3 \times r}$, the corresponding conclusion also holds as long as $\mathcal{X} \times_3 (\mathbf{Q}\mathbf{Q}^\top) = \mathcal{X}$. That is to say, $\|\mathcal{X}\|_{Q,*} \leq \text{rank}_Q(\mathcal{X})$ holds within the region $\{\mathcal{X} \mid \|\mathcal{X}\|_{Q,\sigma} \leq 1\}$.

Theorem 3 shows that though $\|\mathcal{X}\|_{PCA,*}$ could be non-convex, its function value is always below $\text{rank}_{PCA}(\mathcal{X})$. Therefore, model (22) can be regarded as a reasonable low rank tensor recovery model. Notice that it is actually a bilevel optimization problem.

3.2 Way II (MOTQN): estimate Q by manifold optimization

Recalling the data-dependent low rank recovery model Eq. (15) with $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, our main idea is to find a learnable $\mathbf{Q} \in \mathbb{R}^{n_3 \times n_3}$ to minimize $\text{rank}_Q(\mathcal{X})$. Inspired by Remark 3, if we let $\mathbf{Q} = \underset{\mathbf{Q}^\top\mathbf{Q}=\mathbf{I}}{\text{argmin}} \|\mathcal{X}\|_{Q,*}$ to minimize the surrogate function directly, then we can get the following bilevel model:

$$\min_{\mathcal{X},\mathbf{Q}} \|\mathcal{X}\|_{Q,*}, \quad s.t.\ \Psi(\mathcal{X}) = \mathcal{Y},\ \mathbf{Q} = \underset{\mathbf{Q}^\top\mathbf{Q}=\mathbf{I}}{\text{argmin}} \|\mathcal{X}\|_{Q,*}. \tag{26}$$

In Eq. (26), the lower-level problem w.r.t. $\mathbf{Q}$ is actually a Stiefel manifold optimization problem. Similarly, we can define the corresponding nuclear norm as follows:

Definition 6 (*MOTQN: manifold optimization tensor Q-nuclear norm*) Let $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ be a third-order tensor and $\mathbf{Q} \in \mathbb{R}^{n_3 \times n_3}$ be an orthogonal matrix. Then the Manifold Optimization Tensor Q-Nuclear norm (MOTQN) is defined as:

$$\|\mathcal{X}\|_{Q,*}, \quad \text{where } \mathbf{Q} = \underset{\mathbf{Q}^\top\mathbf{Q}=\mathbf{I}}{\text{argmin}} \|\mathcal{X}\|_{Q,*}. \tag{27}$$

Different from VMTQN, the learnable $\mathbf{Q}$ in Eq. (26) should be a square matrix, i.e., $\mathbf{Q} \in \mathbb{R}^{n_3 \times n_3}$. If not, as mentioned in Sect. 3.1.1, $\mathbf{Q}$ may converge to the singular spaces which are corresponding to smaller singular values. To avoid this case, we let $\mathbf{Q} \in \mathbb{R}^{n_3 \times n_3}$. Following, the key point of solving this model is how to deal with such an orthogonality constrained optimization problem:

$$\mathbf{Q} = \underset{\mathbf{Q}^\top\mathbf{Q}=\mathbf{I}}{\text{argmin}} \|\mathcal{X}\|_{Q,*} = \underset{\mathbf{Q}^\top\mathbf{Q}=\mathbf{I}}{\text{argmin}} \sum_{i=1}^{n_3} \left\|\mathbf{G}^{(i)}\right\|_*, \quad \text{where } \mathcal{G} = \mathcal{X} \times_3 \mathbf{Q}. \tag{28}$$

Note that Eq. (28) is actually a non-convex problem due to the orthogonality constraint. The usual way is to perform the manifold Gradient Descent on the Stiefel manifold, which evolves along the manifold geodesics (Edelman et al. 1998). However, this method usually requires a lot of computation to calculate the projected gradient direction of the objective function. Meanwhile, the work Wen and Yin (2013) develops a technique to solve such orthogonality constrained problem **iteratively**, which generates feasible points by the Cayley transformation and only involves matrix multiplication and inversion. Here we consider to use their algorithm to solve the low-level problem.

3.2.1 Optimization with orthogonality constraints

Assume $\mathbf{Q} \in \mathbb{R}^{n\times r}$ and denote the gradient of the objective function $f(\mathbf{Q}) = \|\mathcal{X}\|_{Q,*}$ w.r.t. $\mathbf{Q}$ at $\mathbf{Q}_k$ (the kth iteration) by $\mathbf{P} \in \mathbb{R}^{n\times r}$. Then the projection of $\mathbf{P}$ in the tangent space of the Stiefel manifold at $\mathbf{Q}_k$ is $\mathbf{A}\mathbf{Q}_k$, where $\mathbf{A} = \mathbf{P}\mathbf{Q}_k^\top - \mathbf{Q}_k\mathbf{P}^\top$ and $\mathbf{A} \in \mathbb{R}^{n\times n}$ (Wen and Yin 2013). Instead of parameterizing the geodesic of the Stiefel manifold along direction $\mathbf{A}$ using the exponential function, inspired by Wen and Yin (2013), we generate feasible points by the following Cayley transformation:

$$\mathbf{Q}(\tau) = \mathbf{C}(\tau)\mathbf{Q}_k, \quad \text{where } \mathbf{C}(\tau) = \left(\mathbf{I} + \frac{\tau}{2}\mathbf{A}\right)^{-1}\left(\mathbf{I} - \frac{\tau}{2}\mathbf{A}\right), \tag{29}$$

where $\mathbf{I}$ is the identity matrix and $\tau \in \mathbb{R}$ is a parameter to determine the step size of $\mathbf{Q}_{k+1}$. That is to say, $\mathbf{Q}(\tau)$ is a re-parameterized geodesic w.r.t. τ on the Stiefel manifold. Moreover, if $\mathbf{Q}_k^\top\mathbf{Q}_k = \mathbf{I}$ holds, then $\mathbf{Q}(\tau)$ has the following properties:

(1) $\frac{d}{d\tau}\mathbf{Q}(0) = -\mathbf{A}\mathbf{Q}_k$, (2) $\mathbf{Q}(\tau)$ is smooth in τ, (3) $\mathbf{Q}(0) = \mathbf{Q}_k$, (4) $\mathbf{Q}(\tau)^\top\mathbf{Q}(\tau) = \mathbf{I}$.

The work Wen and Yin (2013) shows that if τ is in a proper range, $\mathbf{Q}(\tau)$ can lead to a lower objective function value than $\mathbf{Q}(0)$ on the Stiefel manifold. In summery, solving the problem $\mathbf{Q} = \underset{\mathbf{Q}^\top\mathbf{Q}=\mathbf{I}}{\operatorname{argmin}} \|\mathcal{X}\|_{Q,*}$ consists of two steps: (1) find a proper τ^* to make the value of the objective function $f(\mathbf{Q}(\tau)) = \|\mathcal{X}\|_{Q(\tau),*}$ decrease; (2) update $\mathbf{Q}_{k+1}$ by Eq. (29), i.e., $\mathbf{Q}_{k+1} = \mathbf{Q}(\tau^*)$.

3.2.2 Details of how to estimate τ^* and update $\mathbf{Q}_k$

(1) We first compute the gradient of the objective function $f(\mathbf{Q}) = \|\mathcal{X}\|_{Q,*}$ w.r.t. $\mathbf{Q}$ at $\mathbf{Q}_k$. According to the chain rule, we get the following:

$$\frac{\partial f(\mathbf{Q})}{\partial \mathbf{Q}} = \frac{\partial \mathcal{G}}{\partial \mathbf{Q}} \cdot \frac{\partial f(\mathbf{Q})}{\partial \mathcal{G}} = \frac{\partial(\mathbf{G}_{(3)})}{\partial \mathbf{Q}} \times \text{unfold}_3\left(\frac{\partial f(\mathbf{Q})}{\partial \mathcal{G}}\right). \tag{30}$$

Note that $\mathcal{G} = \mathcal{X}\times_3 \mathbf{Q}$ and $\mathbf{G}_{(3)} = \mathbf{X}_{(3)}\mathbf{Q}$, then we can get $\frac{\partial \mathbf{G}_{(3)}}{\partial \mathbf{Q}} = \mathbf{X}_{(3)}^\top$ where $\mathbf{G}_{(3)}$ and $\mathbf{X}_{(3)}$ are the mode-3 unfolding matrices. Additionally, Eq. (28) shows that $f(\mathbf{Q}) = \sum_{i=1}^{n_3} \|\mathbf{G}^{(i)}\|_*$ where $\mathbf{G}^{(i)}$ are the frontal slices of $\mathcal{G}$. We let $\mathbf{H}^{(i)} = \mathbf{U}^{(i)}\mathbf{V}^{(i)}$, where $\mathbf{H}^{(i)}$ denotes the frontal slice of $\mathcal{H}$ and $\mathbf{U}^{(i)}\mathbf{V}^{(i)}$ denotes the left/right singular matrices of $\mathbf{G}^{(i)}$ by skinny SVD (Petersen and Pedersen 2008). Therefore, $\mathcal{H} = \frac{\partial f(\mathbf{Q})}{\partial \mathcal{G}}$ is the same as the matrix case and can be obtained from the singular value decomposition.[5]

[5] The subgradient of matrix nuclear norm $\|\mathbf{M}\|_*$ w.r.t. $\mathbf{M}$ is $\{\mathbf{U}\mathbf{V}^\top + \mathbf{W} \parallel \mathbf{U}^\top\mathbf{W} = \mathbf{O}, \mathbf{W}\mathbf{V} = \mathbf{O}, \|\mathbf{W}\| \le 1\}$, where $\mathbf{M} = \mathbf{U}\Sigma\mathbf{V}^\top$ is the SVD of $\mathbf{M}$.

In summary, the gradient of the objective function $f(\mathbf{Q})$ w.r.t. $\mathbf{Q}$ at $\mathbf{Q}_k$ (denoted by $\mathbf{P}$) can be written as follows:

$$\text{Gradient} = \mathbf{P} = \frac{\partial f(\mathbf{Q})}{\partial \mathbf{Q}} = \frac{\partial \mathcal{G}}{\partial \mathbf{Q}} \cdot \frac{\partial f(\mathbf{Q})}{\partial \mathcal{G}} = \mathbf{X}_{(3)}^{\top} \mathbf{H}_{(3)}. \tag{31}$$

where $\mathbf{X}_{(3)}$ and $\mathbf{H}_{(3)}$ are the mode-3 unfolding matrices of $\mathcal{X}$ and $\mathcal{H}$, respectively.

(2) Then we construct a geodesic curve along the gradient direction on the Stiefel manifold by Eq. (29):

$$\mathbf{Q}(\tau) = \left(\mathbf{I} + \frac{\tau}{2}\mathbf{A}\right)^{-1}\left(\mathbf{I} - \frac{\tau}{2}\mathbf{A}\right)\mathbf{Q}_k, \quad \text{where } \mathbf{A} = \mathbf{P}\mathbf{Q}_k^{\top} - \mathbf{Q}_k\mathbf{P}^{\top}. \tag{32}$$

We consider the following problem for finding a proper τ:

$$\tau^* = \underset{0\le\tau\le\epsilon}{\operatorname{argmin}} f(\mathbf{Q}(\tau)) = \underset{0\le\tau\le\epsilon}{\operatorname{argmin}} g(\tau) = \underset{0\le\tau\le\epsilon}{\operatorname{argmin}} \|\mathcal{X}\|_{Q(\tau),*}, \tag{33}$$

where ϵ is a given parameter to ensure that τ^* is small enough and $\|\frac{\tau}{2}\mathbf{A}\| \le 1$ holds. Then we can simplify $g(\tau) = f(\mathbf{Q}(\tau))$ with the equation $\left(\mathbf{I} + \frac{\tau}{2}\mathbf{A}\right)^{-1} = \mathbf{I} + \sum_{l=1}^{\infty}\left(-\frac{\tau}{2}\mathbf{A}\right)^l$ and obtain the following:

$$g(\tau) = f(\mathbf{Q}(\tau)) = f\left(\left(\mathbf{I} + 2\sum_{l=1}^{\infty}\left(-\frac{\tau}{2}\mathbf{A}\right)^l\right)\mathbf{Q}_k\right) \approx f\left(\left(\mathbf{I} - \tau\mathbf{A} + \frac{\tau^2}{2}\mathbf{A}^2\right)\mathbf{Q}_k\right). \tag{34}$$

Given that τ^* is small enough, we can approximate $g(\tau)$ via its second order Taylor expansion at $\tau = 0$, i.e., $g(\tau) = g(0) + g'(0)\cdot\tau + \frac{1}{2}g''(0)\cdot\tau^2 + \mathcal{O}(\tau^3)$. It should be mentioned that since $f(\mathbf{Q})$ is non-convex w.r.t. $\mathbf{Q}$, the sign of $g''(0)$ is uncertain. However, Wen and Yin (2013) point out that $g'(0) = -\frac{1}{2}\|\mathbf{A}\|_F^2$ always holds. Thus we can estimate an optimal solution τ^* via:

$$\tau^* = \min\{\varepsilon, \tilde{\tau}\}, \quad \text{where } \varepsilon < \frac{2}{\|\mathbf{A}\|}, \text{and } \tilde{\tau} = \begin{cases} -\frac{g'(0)}{g''(0)}, & g''(0) > 0 \\ \frac{1}{\|\mathbf{A}\|}, & g''(0) \le 0. \end{cases} \tag{35}$$

Here we give the following lemma to omit the calculation process (see "Appendix D").

Lemma 3 *Let* $g(\tau) = f(\mathbf{Q}(\tau)) = \|\mathcal{X}\|_{Q(\tau),*}$ *and* $\mathbf{Q}(\tau) \approx \left(\mathbf{I} - \tau\mathbf{A} + \frac{\tau^2}{2}\mathbf{A}^2\right)\mathbf{Q}_k$, *where* $\mathbf{A}$ *is defined in Eq.* (32). *Then the first and the second order derivatives of* $g(\tau)$ *evaluated at* 0 *can be estimated as follows*:

$$g'(0) \approx \left\langle \mathbf{X}_{(3)}^{\top}\mathbf{H}_{(3)}, -\mathbf{A}\mathbf{Q}_k \right\rangle, \text{ and } g''(0) \approx \left\langle \mathbf{X}_{(3)}^{\top}\mathbf{H}_{(3)}, \mathbf{A}^2\mathbf{Q}_k \right\rangle, \tag{36}$$

where $\mathbf{X}_{(3)}$ *and* $\mathbf{H}_{(3)}$ *are defined as the same in Eq.* (31).

By using Eq. (35) and Lemma 3, we can obtain the optimal step size τ^* and then use Eq. (32) to update $\mathbf{Q}_{k+1} = \mathbf{Q}(\tau^*)$. Algorithm 1 organizes the whole calculation process.

Back to the bilevel low rank tensor recovery model Eq. (26), for the lower-level problem Eq. (28), we finish the iterative updating step by Algorithm 1. Once $\mathbf{Q}_{k+1}$ is fixed, the upper-level problem can be solved easily.

Algorithm 1 Updating $\mathbf{Q}$ iteratively to solve Eq. (27).

Input: Tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, orthogonal matrix $\mathbf{Q_0} \in \mathbb{R}^{n_3 \times n_3}$.

1. **while not convergence**
2. Calculate $\mathbf{P} = \mathbf{X}_{(3)}^{\top}\mathbf{H}_{(3)}$ by Eq. (31).
3. Calculate $\mathbf{A} = \mathbf{P}\mathbf{Q}_k^{\top} - \mathbf{Q}_k\mathbf{P}^{\top}$ by Eq. (32).
4. Estimate $\tau^* = \min\{\varepsilon, \tilde{\tau}\}$ by Eq. (35) and Lemma 3.
5. Update $\mathbf{Q}_{k+1} = \mathbf{Q}(\tau^*)$ by Eq. (32).
6. **end while**

Output: Matrix $\mathbf{Q}_K$.

3.3 Applicability of VMTQN and MOTQN

In Sect. 3.2 (MOTQN), we mention that $\mathbf{Q} \in \mathbb{R}^{n_3 \times n_3}$ should be a square matrix but not in Sect. 3.1 (VMTQN). In this section, we start from this point and analyze the impact of the size of $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ on the applicability of these two methods.

3.3.1 Case 1: $r = n_1 n_2 \ll n_3$

In this case, VMTQN model in Eq. (22) usually performs better than other methods in terms of computational efficiency, including MOTQN and other works (Zhang and Aeron 2017; Xu et al. 2019; Song et al. 2019; Lu et al. 2019; Jiang et al. 2020). As we can see from Sect. 3.1 of VMTQN model, we need to calculate a skinny right singular matrix $\mathbf{V}$ of an unfolding matrix $\mathbf{X}_{(3)} \in \mathbb{R}^{n_1 n_2 \times n_3}$. If $r < n_3$, then not only the computational complexity is not too large, but $\mathbf{Q}$ can play the role of feature selection like Principal Component Analysis, which corresponds to the notation $\mathbf{Q} = \text{PCA}(\mathcal{X}, 3, r)$.

Meanwhile, MOTQN and the work Zhang and Aeron (2017),Xu et al. (2019), Song et al. (2019) and Lu et al. (2019) usually need to have a square factor matrix $\mathbf{Q}$, even that (Jiang et al. 2020) requires the columns of $\mathbf{Q}$ to be redundant.

3.3.2 Case 2: $n_1 n_2 > n_3 = r$ or even have the same order of magnitude

In this case, MOTQN model in Eq. (26) has the best explainability and rationality. On the one hand, with the same size of $\mathbf{Q} \in \mathbb{R}^{n_3 \times n_3}$, MOTQN minimize the tensor Q-nuclear norm directly, which corresponds to the definition of low rank structure properly. On the other hand, thanks to the algorithm in Wen and Yin (2013), the optimization of MOTQN model has good convergence guarantee.

4 Applications to tensor completion

4.1 Low rank tensor completion model

In the third-order tensor tensor completion task, Ω is an index set consisting of the indices $\{(i, j, k)\}$ which can be observed, and the operator Ψ in Eqs. (21) and (22) is replaced by an

orthogonal projection operator $\mathcal{P}_\Omega$, where $\mathcal{P}_\Omega(\mathcal{X}_{ijk}) = \mathcal{X}_{ijk}$ if $(i,j,k) \in \Omega$ and 0 otherwise. The observed tensor $\mathcal{Y}$ satisfies $\mathcal{Y} = \mathcal{P}_\Omega(\mathcal{Y})$. Then the tensor completion model based on our two ways are given by:

$$(VMTQN): \quad \min_{\mathcal{X}} \|\mathcal{X}\|_{Q,*}, \quad s.t.\ \mathcal{P}_\Omega(\mathcal{X}) = \mathcal{Y},\ \mathbf{Q} = \mathrm{PCA}(\mathcal{X}, 3, r), \tag{37}$$

and

$$\begin{aligned}(MOTQN): \quad & \min_{\mathcal{X},\mathbf{Q}} \|\mathcal{X}\|_{Q,*} \\ & s.t.\ \mathbf{Q} = \operatorname*{argmin}_{\mathbf{Q}^\top\mathbf{Q}=\mathbf{I}} \|\mathcal{X}\|_{Q,*},\ \mathcal{P}_\Omega(\mathcal{X}) = \mathcal{Y}.\end{aligned} \tag{38}$$

where $\mathcal{X}$ is the tensor that has low rank structure. In Eq. (37), $\mathbf{Q} \in \mathbb{R}^{n_3 \times r}$ is an column orthonormal matrix with $r = \min\{n_1 n_2, n_3\}$. While in Eq. (38), $\mathbf{Q} \in \mathbb{R}^{n_3 \times n_3}$ is a square orthogonal matrix. To solve these models by ADMM based method (Lu et al. 2017), we introduce an intermediate tensor $\mathcal{E}$ to separate $\mathcal{X}$ from $\mathcal{P}_\Omega(\cdot)$. Let $\mathcal{E} = \mathcal{P}_\Omega(\mathcal{X}) - \mathcal{X}$, then $\mathcal{P}_\Omega(\mathcal{X}) = \mathcal{Y}$ is translated to $\mathcal{X} + \mathcal{E} = \mathcal{Y},\ \mathcal{P}_\Omega(\mathcal{E}) = \mathcal{O}$, where $\mathcal{O}$ is an all-zero tensor. Then we get the following two models:

$$(VMTQN): \ \min_{\mathcal{X},\mathcal{E},\mathbf{Q}} \|\mathcal{X}\|_{Q,*},\ s.t.\ \mathcal{X} + \mathcal{E} = \mathcal{Y},\ \mathcal{P}_\Omega(\mathcal{E}) = \mathcal{O},\ \mathbf{Q} = \mathrm{PCA}(\mathcal{X}, 3, r), \tag{39}$$

and

$$(MOTQN): \ \min_{\mathcal{X},\mathcal{E},\mathbf{Q}} \|\mathcal{X}\|_{Q,*},\ s.t.\ \mathcal{X} + \mathcal{E} = \mathcal{Y},\ \mathcal{P}_\Omega(\mathcal{E}) = \mathcal{O},\ \mathbf{Q} = \operatorname*{argmin}_{\mathbf{Q}^\top\mathbf{Q}=\mathbf{I}} \|\mathcal{X}\|_{Q,*}. \tag{40}$$

Note that in Eq. (40), the constraint $\mathbf{Q} = \operatorname*{argmin}_{\mathbf{Q}^\top\mathbf{Q}=\mathbf{I}} \|\mathcal{X}\|_{Q,*}$ is the same as the objective function, thus it can be omitted. Nevertheless, in order to keep Eqs. (39) and (40) unified in form and express the dependence of $\mathbf{Q}$ and $\mathcal{X}$ conveniently, we reserve this constraint here.

4.2 Optimization algorithm

Since $\mathbf{Q}$ is dependent on $\mathcal{X}$, it is difficult to solve the models (39) and (40) w.r.t. $\{\mathcal{X}, \mathbf{Q}\}$ directly. Here we adopt the idea of alternating minimization to solve $\mathcal{X}$ and $\mathbf{Q}$ alternately. We separate the sub-problem of solving $\mathbf{Q}$ as a sub-step in every K-iteration, and then update $\mathcal{X}$ with a fixed $\mathbf{Q}$ by the ADMM method (Lu et al. 2017, 2018). The partial augmented Lagrangian function of Eqs. (39) and (40) is

$$L(\mathcal{X}, \mathcal{E}, \mathcal{Z}, \mu) = \|\mathcal{X}\|_{Q,*} + \langle \mathcal{Z}, \mathcal{Y} - \mathcal{X} - \mathcal{E} \rangle + \frac{\mu}{2}\|\mathcal{Y} - \mathcal{X} - \mathcal{E}\|_F^2, \tag{41}$$

where $\mathcal{Z}$ is the dual variable and $\mu > 0$ is the penalty parameter. Then we can update each component $\mathbf{Q}$, $\mathcal{X}$, $\mathcal{E}$, and $\mathcal{Z}$ alternately. Algorithms 2 and 3 show the details about the optimization methods to Eqs. (39) and (40). In order to improve the efficiency and stable convergence of the algorithm, we introduce a parameter K to control the update frequency of $\mathbf{Q}$ with the help of heuristic design. The different effects of K on the two models are explained in Sect. 4.3 and Sect. 4.4, respectively.

Note that there is one operator **Prox** in the sub-step of updating $\mathcal{X}$ as follows:

$$\mathcal{X} = \mathbf{Prox}_{\lambda\|\cdot\|_{Q,*}}(\mathcal{T}) := \underset{\mathcal{X}}{\operatorname{argmin}}\ \lambda\|\mathcal{X}\|_{Q,*} + \frac{1}{2}\|\mathcal{X}-\mathcal{T}\|_F^2, \tag{42}$$

where $\mathbf{Q} \in \mathbb{R}^{n_3\times r}$ is a given column orthonormal matrix and $\|\mathcal{X}\|_{Q,*}$ is the tensor Q-nuclear norm of $\mathcal{X}$ which is defined in Eq. (12). Algorithm 3 shows the details of solving this operator.

Algorithm 2 Solving the problems (39) and (40):VMTQN and MOTQN models by ADMM.

Input: Observation samples $\mathcal{Y}_{ijk}$, $(i,j,k) \in \Omega$, of tensor $\mathcal{Y} \in \mathbb{R}^{n_1\times n_2\times n_3}$.
Initialize: $\mathcal{X}_0,\ \mathcal{E}_0,\ \mathcal{Z}_0,\ \mathbf{Q}_0 \in \mathbb{R}^{n_3\times r}$. Parameters $k=1,\ \rho>1,\ \mu_0,\ \mu_{max},\ \varepsilon,\ K$.
While not converge **do**

1. Update $\mathbf{Q}_k$ by one of the following:

$$\textbf{(VMTQN):}\quad \mathbf{Q}_k = \begin{cases} \mathbf{Q}_{k-1}, & k \bmod K \neq K-1, \\ \mathrm{PCA}\left(\mathcal{Y}-\mathcal{E}_{k-1}+\frac{\mathcal{Z}_{k-1}}{\mu_{k-1}}, 3, r\right), & k \bmod K = K-1. \end{cases} \tag{43}$$

$$\textbf{(MOTQN):}\quad \mathbf{Q}_k = \begin{cases} \mathbf{Q}_{k-1}, & k \bmod K \neq K-1, \\ \mathbf{Q}(\tau^*) \quad \text{by using Algorithm 1}, & k \bmod K = K-1. \end{cases} \tag{44}$$

2. Update $\mathcal{X}_k$ by

$$\mathcal{X}_k = \mathbf{Prox}_{\mu_{k-1}^{-1}\|\cdot\|_{Q_k,*}}\left(\mathcal{Y}-\mathcal{E}_{k-1}+\frac{\mathcal{Z}_{k-1}}{\mu_{k-1}}\right). \tag{45}$$

3. Update $\mathcal{E}_k$ by

$$\mathcal{E}_k = \mathcal{P}_{\Omega^{\complement}}\left(\mathcal{Y}-\mathcal{X}_k+\frac{\mathcal{Z}_{k-1}}{\mu_{k-1}}\right), \tag{46}$$

where $\Omega^{\complement}$ is the complement of Ω.

4. Update the dual variable $\mathcal{Z}_k$ by

$$\mathcal{Z}_k = \mathcal{Z}_{k-1} + \mu_{k-1}\left(\mathcal{Y}-\mathcal{X}_k-\mathcal{E}_k\right). \tag{47}$$

5. Update μ_k by

$$\mu_k = \min\{\rho\mu_{k-1}, \mu_{max}\}. \tag{48}$$

6. Check the convergence condition: $\|\mathcal{X}_k-\mathcal{X}_{k-1}\|_\infty \le \varepsilon$, $\|\mathcal{E}_k-\mathcal{E}_{k-1}\|_\infty \le \varepsilon$, and $\|\mathcal{Y}-\mathcal{X}_k-\mathcal{E}_k\|_\infty \le \varepsilon$.
7. $k \leftarrow k+1$.

end While
Output: The target tensor $\mathcal{X}_k$.

Algorithm 3 Solving the proximal operator $\mathbf{Prox}_{\lambda,\|\cdot\|_{Q,*}}(\mathcal{T})$ in Eq. (42) and (45).

Input: Tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, column orthonormal matrix $\mathbf{Q} \in \mathbb{R}^{n_3 \times r}$.

1. $\mathcal{G} = \mathcal{T} \times_3 \mathbf{Q}$.
2. **for** $i = 1$ to r:
   $[\mathbf{U}, \mathbf{S}, \mathbf{V}] = \mathrm{SVD}(\mathbf{G}^{(i)})$.
   $\mathbf{G}^{(i)} = \mathbf{U}(\mathbf{S} - \lambda \mathbf{I})_+ \mathbf{V}^\top$, where $(x)_+ = \max\{x, 0\}$.
3. **end for**
4. $\mathcal{X} = \mathcal{G} \times_3 \mathbf{Q}^\top + \mathcal{T} \times_3 (\mathbf{I} - \mathbf{Q}\mathbf{Q}^\top)$.

Output: Tensor $\mathcal{X}$.

4.3 Convergence analysis

4.3.1 VMTQN model

For the models (37) or (39), it is hard to analyze the convergence of the corresponding optimization method directly. The constraint on $\mathbf{Q}$ is non-linear and the objective function is essentially non-convex w.r.t. $\mathbf{Q}$, which increase the difficulty of analysis. However, the conclusions of Lu et al. (2017), Lin et al. (2015), Xu and Yin (2015), Lin et al. (2011) and Absil et al. (2009) guarantee the convergence to some extent.

In practical applications, we can fix $\mathbf{Q}_k = \mathbf{Q}$ in every K iterations to solve a convex problem w.r.t. $\mathcal{X}$. As long as $\mathcal{X}$ is convergent, by using the following Lemma 4, the change of $\mathbf{Q}$ is bounded.

Lemma 4 (Petersen and Pedersen 2008) *Given a matrix* $\mathbf{X}$ *and its Singular Value Decomposition* $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$. *Let* $\mathbf{v}_i$ *denotes the* ith *column of matrix* $\mathbf{V}$ *and* σ_j *denotes the* j*th singular value of matrix* $\mathbf{X}$. *Denote the sub-differential of a variable by* $\partial(\cdot)$, *then we have the following*:

$$\partial(\mathbf{v}_i) = \left(\sigma_i^2 \mathbf{I} - \mathbf{X}^\top \mathbf{X}\right)^+ \partial(\mathbf{X}^\top \mathbf{X})\mathbf{v}_i. \tag{49}$$

If v_{ij} *represents the* j*th element of* $\mathbf{v}_i$, *then* $\left\| \frac{\partial(v_{ij})}{\partial(\mathbf{X}^\top \mathbf{X})} \right\|_2 < \infty$.

Lemma 4 indicates that, as long as the change of $\mathcal{X}$ is bounded by penalty term with proper K and ρ, the change of $\mathbf{Q}$ will also be bounded to some extent. Then $\lim_{k\to\infty} Q_k \approx \mathrm{PCA}(\mathcal{X}_k, 3, r)$ gradually meets the constraints.

Unfortunately, Updating the variable $\mathbf{Q}_k$ in Eq. (43) needs to solve a singular linear system, while the objective norm $\|\mathcal{X}\|_{Q,*}$ in Eq. (39) is non-convex w.r.t. $\mathbf{Q}$. Therefore, it is difficult to prove the conclusion that the Lagrangian function in Eq. (41) of Algorithm 2 decreases strictly in each iteration. However, we give another Theorem that the iterations corresponding to Eqs. (45)–(48) are convergent in the case of fixed $\mathbf{Q}$.

Theorem 4 *Given a fixed* $\mathbf{Q}$ *in every* K *iterations, the tensor completion model* (39) *can be solved effectively by Algorithm* 2 *with* $\mathbf{Q}_k = \mathbf{Q}$ *in Eq.* (43), *where* Ψ *is replaced by* $\mathcal{P}_\Omega$. *The rigorous convergence guarantees can be obtained directly due to the convexity as follows*.

Let $(\mathcal{X}^*, \mathcal{E}^*, \mathcal{Z}^*)$ *be one KKT point of problem* (39) *with fixed* $\mathbf{Q}$, $\hat{\mathcal{X}}_K = \frac{\sum_{k=0}^{K} \frac{1}{\mu_k} \mathcal{X}_{k+1}}{\sum_{k=0}^{K} \frac{1}{\mu_k}}$, *and* $\hat{\mathcal{E}}_K = \frac{\sum_{k=0}^{K} \frac{1}{\mu_k} \mathcal{E}_{k+1}}{\sum_{k=0}^{K} \frac{1}{\mu_k}}$, *then we have*

$$\|\hat{\mathcal{X}}_{K+1} + \hat{\mathcal{E}}_{K+1} - \mathcal{Y}\|_F^2 \leq O\left(\frac{1}{\sum_{k=0}^{K} \frac{1}{\mu_k}}\right), \tag{50}$$

and

$$0 \leq \|\hat{\mathcal{X}}_{K+1}\|_{Q,*} - \|\mathcal{X}^*\|_{Q,*} + \left\langle \mathcal{Z}^*, \hat{\mathcal{X}}_{K+1} + \hat{\mathcal{E}}_{K+1} - \mathcal{Y} \right\rangle \leq O\left(\frac{1}{\sum_{k=0}^{K} \frac{1}{\mu_k}}\right). \tag{51}$$

4.3.2 MOTQN model

Different from VMTQN model, as we mentioned in Sect. 3.3.2, MOTQN model has a complete guarantee of convergence with the help of Wen and Yin (2013). The updating step in Eq. (44) can strictly guarantee the decrease of the objective function value $\|\mathcal{X}\|_{Q,*}$ with a proper step size τ^*.

Lemma 5 (Lemma 3 of Wen and Yin (2013)) *Denote the gradient of the objective function* $f(\mathbf{Q})$ *w.r.t.* $\mathbf{Q}$ *at* $\mathbf{Q}_k$ *by* $\mathbf{P}$ *and let* $\mathbf{A} = \mathbf{P}\mathbf{Q}_k^\top - \mathbf{Q}_k\mathbf{P}^\top$ *be a skew-symmetric matrix. If we define* $\mathbf{Q}(\tau)$ *by Eq.* (32), *then* $\mathbf{Q}(\tau)$ *is a descent curve at* $\tau = 0$, *that is*,

$$f'_\tau(\mathbf{Q}(0)) := \frac{\partial f(\mathbf{Q}(\tau))}{\partial \tau}\Big|_{\tau=0} = -\frac{1}{2}\|\mathbf{A}\|_F^2 \leq 0. \tag{52}$$

Lemma 5 indicates that, as long as τ is small enough, Eq. (44) usually decreases the value of $f(\mathbf{Q}(\tau))$. Notice that Eq. (41) is a partial augmented Lagrangian function, hence the value of Lagrangian function will also decreases after Eq. (44). Therefore, we have the following theorem to ensure the convergence of Algorithm 2:

Theorem 5 *Denote the augmented Lagrangian function of low rank tensor recovery model* (38) *by* $L(\mathbf{Q}, \mathcal{X}, \mathcal{E}, \mathcal{Z}, \mu)$, *which is shown as follows*:

$$L(\mathbf{Q}, \mathcal{X}, \mathcal{E}, \mathcal{Z}, \mu) = \|\mathcal{X}\|_{Q,*} + \langle \mathcal{Z}, \mathcal{Y} - \mathcal{X} - \mathcal{E} \rangle + \frac{\mu}{2}\|\mathcal{Y} - \mathcal{X} - \mathcal{E}\|_F^2. \tag{53}$$

Then the sequence $\{\mathbf{Q}_k, \mathcal{X}_k, \mathcal{E}_k, \mathcal{Z}_k, \mu_k\}$ *generated in Algorithm* 2 *with Eq.* (44) *satisfies the following*:

$$\begin{aligned} L(\mathbf{Q}_k, \mathcal{X}_k, \mathcal{E}_k, \mathcal{Z}_k, \mu_k) &\geq L(\mathbf{Q}_{k+1}, \mathcal{X}_{k+1}, \mathcal{E}_{k+1}, \mathcal{Z}_{k+1}, \mu_{k+1}) \\ &+ \frac{\mu_k}{2}\left\|\mathcal{E}_k - \mathcal{E}_{k+1}\right\|_F^2 + \left(\frac{\mu_k}{2} - \frac{\mu_{k+1} + \mu_k}{2\mu_k^2} C_L\right)\left\|\mathcal{X}_k - \mathcal{X}_{k+1}\right\|_F^2. \end{aligned} \tag{54}$$

The function value of Eq. (53) *decreases monotonically after each iteration as long as* $\mu \geq \sqrt{(\rho+1)C_L}$, *where* ρ *is defined in Eq.* (48) *and* C_L *is a constant w.r.t.* $\mathcal{X}$. *By the monotone bounded convergence theorem, Algorithm* 2 *is convergent.*

4.4 Complexity analysis

The computational complexity of VMTQN in Eq. (43) is $\mathcal{O}\big(rn_1n_2n_3\big)$, where r denotes the number of columns of $\mathbf{Q} \in \mathbb{R}^{n_3 \times r}$. And the complexity of MOTQN in Eq. (44) is $\mathcal{O}\big((n_1n_2 + n_3)n_3^2\big)$. As for TNN based method Zhang et al. (2014); Zhang and Aeron (2017); Lu et al. (2016) and Lu et al. (2018), they use Fourier transform and have a complexity of $\mathcal{O}\big(n_1n_2n_3 \log n_3\big)$. As can be seen, if $r < \log n_3$, VMTQN can be more efficient than the other two methods. Otherwise, we should use a larger K to control the overall calculation speed.

However, when solving our two methods or TNN based method, the most time-consuming part is in the SVD operator of each iteration, which is corresponding to Eqs. (45)–(48). In this part, VMTQN based method has a complexity of $\mathcal{O}\big(rn_1n_2 \min\{n_1, n_2\}\big)$ while MOTQN and TNN based methods has a complexity of $\mathcal{O}\big(n_3n_1n_2 \min\{n_1, n_2\}\big)$. That is to say, as long as $r \ll n_3$, VMTQN based method is usually more efficient than the other two methods.

4.5 Performance analysis

Considering the low rank tensor recovery models in Eqs. (37) and (38), Ω is an index set consisting of the indices $\{(i, j, k)\}$ which can be observed, and the orthogonal projection operator $\mathcal{P}_\Omega$ is defined as $\mathcal{P}_\Omega(\mathcal{X}_{ijk}) = \mathcal{X}_{ijk}$ if $(i, j, k) \in \Omega$ and 0 otherwise. In this part, we discuss at least how many observation samples $|\Omega|$ are needed to recover the ground-truth. In fact, $\mathbf{Q}^*$ obtained from the convergence of Algorithm 2 has a decisive effect on the number of observation samples needed, since the optimal solution satisfies the KKT conditions under $\mathbf{Q}^*$. That is to say, we only need to analyze the performance guarantee in the case of fixed $\mathbf{Q}$.

With a fixed $\mathbf{Q}$, the exact tensor completion guarantee for model (13) is shown in Theorem 6. Lu et al. (2019) also have similar conclusions.

Theorem 6 *Given a fixed orthogonal matrix $\mathbf{Q} \in \mathbb{R}^{n_3 \times n_3}$ and $\Omega \sim Ber(p)$, assume that tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ $(n_1 \geq n_2)$ has a low tensor Q-rank structure and $\text{rank}_Q(\mathcal{X}) = R$. If $|\Omega| \geq \mathcal{O}(\mu R n_1 \log(n_1 n_3))$, then $\mathcal{X}$ is the unique solution to Eq. (13) with high probability, where Ψ is replaced by $\mathcal{P}_\Omega$, and μ is the corresponding incoherence parameter (see Supplementary Materials).*

Through the proof of Lu et al. (2019) and Lu et al. (2018), the sampling rate p should be proportional to $\max\{\|\mathcal{P}_\mathcal{T}(\mathbf{e}_{ijk})\|_F^2\}$. (The definition of projection operators $\mathcal{P}_\mathcal{T}$ and $\mathbf{e}_{ijk}$ can be found in Lu et al. (2016) and Lu et al. (2018) or in Supplementary materials, where $\mathcal{T}$ is the singular space of the ground-truth.) The projection of $\mathbf{e}_{ijk}$ onto subspace $\mathcal{T}$ is greatly influenced by the dimension. Obviously, when $\mathcal{T}$ is the whole space, $\left\|\mathcal{P}_{\mathcal{T}_Q}(\mathbf{e}_{ijk})\right\|_F^2 = 1$. That is to say, a small dimension of $\mathcal{T}_Q$ may lead to a small $\max_{ijk}\left\{\left\|\mathcal{P}_{\mathcal{T}_Q}(\mathbf{e}_{ijk})\right\|_F^2\right\}$.

Proposition 15 in Lu et al. (2018) also implies that for any $\Delta \in \mathcal{T}$, we need to have $\mathcal{P}_\Omega(\Delta) = 0 \Leftrightarrow \Delta = 0$. These two conditions indicate that once the spatial dimension of $\mathcal{T}$

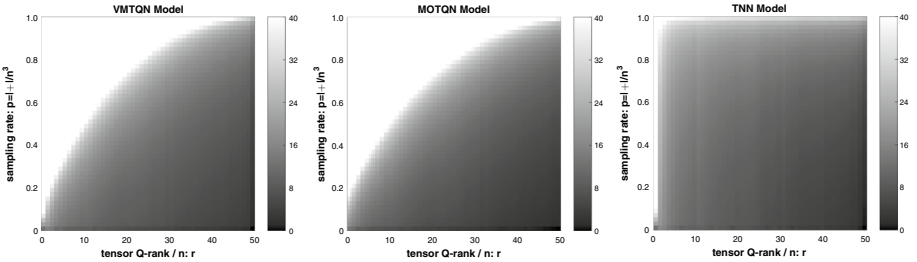

Fig. 3 The numbers plotted on the above figure are the average PSNRs within 10 random trials. The gray scale reflects the quality of completion results of three different models (VMTQN, MOTQN, TNN), where $n_1 = n_2 = n_3 = 50$ and the white area represents a maximum PSNR of 40

Table 1 Comparisons of PSNR results on CIFAR images with different sampling rates

Sampling rate p	0.1	0.2	0.3	0.4	0.5	0.6
TQN with random **Q**	10.86	15.47	18.09	20.20	22.30	24.49
TQN with Oracle **Q** (ideal)	25.39	30.85	39.43	**109.52**	**>200**	**>200**
VMTQN (ours)	**18.83**	**21.10**	**22.89**	**24.56**	**26.26**	**28.07**
TNN (Fourier) Lu et al. (2018)	9.84	12.73	15.68	18.71	21.60	24.26
TNN-C (cosine) Xu et al. (2019)	9.63	11.92	15.17	18.45	22.09	23.95
TTNN (wavelet) Song et al. (2019)	8.97	13.08	17.19	19.26	23.13	25.67
F-TNN (framelet) Jiang et al. (2020)	8.84	11.95	16.56	20.61	23.77	26.02
Tmac Xu et al. (2017)	17.81	19.29	23.06	24.89	25.74	27.46
SiLRTC Liu et al. (2013)	16.87	20.04	21.99	23.80	25.62	27.57
Sampling rate p	0.1	0.2	0.3	0.4	0.5	0.6
TQN with random **Q**	10.84	15.45	18.06	20.19	22.29	24.48
TQN with Oracle **Q** (ideal)	45.75	**>200**	**>200**	**>200**	**>200**	**>200**
VMTQN (ours)	**19.06**	**21.43**	**23.27**	**24.97**	**26.65**	**28.42**
TNN (Fourier) Lu et al. (2018)	8.18	10.10	12.19	14.63	17.59	21.20
TNN-C (cosine) Xu et al. (2019)	8.12	9.95	11.80	13.62	18.07	22.10
TTNN (wavelet) Song et al. (2019)	9.01	10.80	13.27	15.88	20.21	24.04
F-TNN (framelet) Jiang et al. (2020)	9.17	11.06	15.10	17.44	20.85	23.77
Tmac Xu et al. (2017)	12.91	18.49	22.97	25.25	27.06	27.97
SiLRTC Liu et al. (2013)	14.02	19.65	22.44	24.38	26.21	28.12

Top: experiments on the case $\mathcal{Y}_1 \in \mathbb{R}^{32\times32\times3000}$. Bottom: experiments on the case $\mathcal{Y}_2 \in \mathbb{R}^{32\times32\times10,000}$

is large ($\text{rank}_Q(\mathcal{X}) = R$ is large), a larger sampling rate p is needed. And Fig. 3 in Lu et al. (2018) verifies the rationality of this deduction by experiment.

In fact, the smoothness of data along the third dimension has a great influence on the Dimension of Freedom (DoF) of space $\mathcal{T}$. Non-smooth change along the third dimension is likely to increase the spatial dimension of $\mathcal{T}$ under the Fourier basis vectors, which makes the TNN based methods ineffective. Our experiments on CIFAR-10 (Table 1) confirm this conclusion.

As for the models (39) and (40) with adaptive $\mathbf{Q}$, our motivation is to find a better $\mathbf{Q}$ in order to make $\text{rank}_Q(\mathcal{X}) = R$ smaller and make the spatial dimension of corresponding $\mathcal{T}_Q$ as small as possible, where $\mathcal{T}_Q$ is the singular space of the ground-truth under $\mathbf{Q}$. In other words, for more complex data with non-smoothness along the third dimension, the adaptive $\mathbf{Q}$ may reduce the dimension of $\mathcal{T}_Q$ and make $\max\{\|\mathcal{P}_{\mathcal{T}_Q}(\mathbf{e}_{ijk})\|_F^2\}$ smaller than $\max\{\|\mathcal{P}_{\mathcal{T}}(\mathbf{e}_{ijk})\|_F^2\}$, leading to a lower bound for the sampling rate p.

5 Experiments

In this section, we conduct numerical experiments to evaluate our proposed models (39) and (40). The platform is Matlab R2018b under Windows 10 on a PC with an Intel i5-7500 CPU and 16 GB memory. The experimental code of most comparison methods comes from the released version. As for some methods without released code, we reproduce it in Matlab 2018b strictly according to the algorithm in their respective papers.

Assume that the observed corrupted tensor is $\mathcal{Y}$, and the true tensor is $\mathcal{X}_0 \in \mathbb{R}^{n_1 \times n_2 \times n_3}$. We represent the recovered tensor (output of the algorithms) as $\mathcal{X}$, and use Peak Signal-to-Noise Ratio (PSNR) to measure the reconstruction error:

$$\text{PSNR} = 10 \log_{10} \left(\frac{n_1 n_2 n_3 \|\mathcal{X}_0\|_\infty^2}{\|\mathcal{X} - \mathcal{X}_0\|_F^2} \right). \tag{55}$$

5.1 Synthetic experiments

In this part we compare our proposed methods (named VMTQN model and MOTQN model) with the mainstream algorithm TNN (Zhang et al. 2014; Lu et al. 2018).

We examine the completion task with varying tensor Q-rank of tensor $\mathcal{Y}$ and varying sampling rate p. Firstly, we generate a random tensor $\mathcal{M} \in \mathbb{R}^{50\times50\times50}$, whose entries are independently sampled from an $\mathcal{N}(0, 1/50)$ distribution. Actually, the data generated in this way is usually non-smooth along each dimension. Then we choose p in $[0.01 : 0.02 : 0.99]$ and r in $[1 : 1 : 50]$, where the column orthonormal matrix $\mathbf{W} \in \mathbb{R}^{50\times r}$ satisfies $\mathbf{W} = \text{PCA}(\mathcal{M}, 3, r)$. We let $\mathcal{Y} = \mathcal{M} \times_3 \mathbf{W} \times_3 \mathbf{W}^\top$ be the true tensor. After that, we create the index set Ω by using a Bernoulli model to randomly sample a subset from $\{1, \ldots, 50\} \times \{1, \ldots, 50\} \times \{1, \ldots, 50\}$. The sampling rate p is $|\Omega|/50^3$. For each pair of (p, r), we simulate 10 times with different random seeds and take the average as the final result. As for the parameters of VMTQN and MOTQN models in Algorithm 2, we set $\rho = 1.1$, $\mu_0 = 10^{-4}$, $\mu_{max} = 10^{10}$, and $\epsilon = 10^{-8}$.

As shown in the upper left corner regions of VMTQN model and MOTQN model in Fig. 3, Algorithm 2 can effectively solve our proposed recovery models (37) and (38). The larger tensor Q-rank it is, the larger the sampling rate p is needed, which is consistent with our Performance Analysis in Theorem 6. By comparing the results of three methods, we can find that TNN has very poor robustness to the data with non-smooth change. And the results of the left and middle images demonstrate our assumptions (Motivation), which may imply that better low rank structure leads to better recovery.

5.2 Real-world datasets

In this part we compare our proposed method with TNN (Lu et al. 2018) with Fourier matrix, TTNN (Song et al. 2019) with wavelet matrix, TNN-C (Xu et al. 2019) with cosine matrix, F-TNN (Jiang et al. 2020) with framelet matrix, SiLRTC (Liu et al. 2013), Tmac (Xu et al. 2017), and Latent Trace Norm (Tomioka and Suzuki 2013). We validate our algorithm on three datasets: (1) CIFAR-10;[6] (2) COIL-20;[7] (3) HMDB51.[8] We set $\rho = 1.1$, $\mu_0 = 10^{-4}$, $\mu_{max} = 10^{10}$, $\epsilon = 10^{-8}$, and $K = 1$ in our methods. As for TNN, SiLRTC, Tmac, F-TNN, and Latent Trace Norm, we use the default settings as in their released code, e.g., Lu et al.[9] and Tomioka et al.[10] For TTNN and TNN-C of unreleased code, we implement their algorithms in MATLAB strictly according to the corresponding papers.

5.2.1 Influences of Q

Corresponding to our motivation, we use a Random orthogonal matrix and an Oracle matrix (the matrix of right singular vectors of the ground-truth unfolding matrix) to test the influence of $\mathbf{Q}$. The results of TQN models with different orthogonal matrix in Tables 1 and 2 show that $\mathbf{Q}$ play an important role in tensor recovery, where the best recovery results among the comparison methods are marked in bold. Comparing with Random $\mathbf{Q}$ case, our Algorithm 2 is effective for searching a better $\mathbf{Q}$. Table 1 also shows that a proper $\mathbf{Q}$ may make recover the ground-truth more easily. For example, with sampling rate $p \geq 0.2$ on 10, 000 images, an Oracle matrix $\mathbf{Q}$ can lead to an "exact" recovery (PSNR > 200).

5.2.2 CIFAR-10

We consider the worst case for TNN based methods that there is almost no smoothness along the third dimension of the data. We randomly selected 3000 and 10,000 images from one batch of CIFAR-10 (Krizhevsky and Hinton 2009) as our true tensors $\mathcal{Y}_1 \in \mathbb{R}^{32\times32\times3000}$ and $\mathcal{Y}_2 \in \mathbb{R}^{32\times32\times10,000}$, respectively. Then we solve the model (39) with our proposed Algorithm 2. The results are shown in Table 1. Note that in the latter case $r = n_1 n_2 \ll n_3$ holds, MOTQN model has high computational complexity. Thus we will not compare it in this part.

Table 1 verifies our hypothesis that TNN regularization performs badly on data with non-smooth change along the third dimension. Our VMTQN method is obviously better than the other methods in the case of low sampling rate. Moreover, by comparing the two groups of experiments, we can see that VMTQN, TMac, and SiLRTC perform better in $\mathcal{Y}_2$. This may be due to that increasing the data volume will make the principal components more significant. Meanwhile, in the methods of Fourier matrix, cosine matrix and wavelet matrix, they almost have no recovery effect when the sampling rate p is lower. This indicates that these specified projection bases can not learn the data features in the case of poor continuity and insufficient sampling.

[6] http://www.cs.toronto.edu/~kriz/cifar.html.
[7] http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php.
[8] http://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/.
[9] https://github.com/canyilu/LibADMM.
[10] https://github.com/ryotat/tensor.

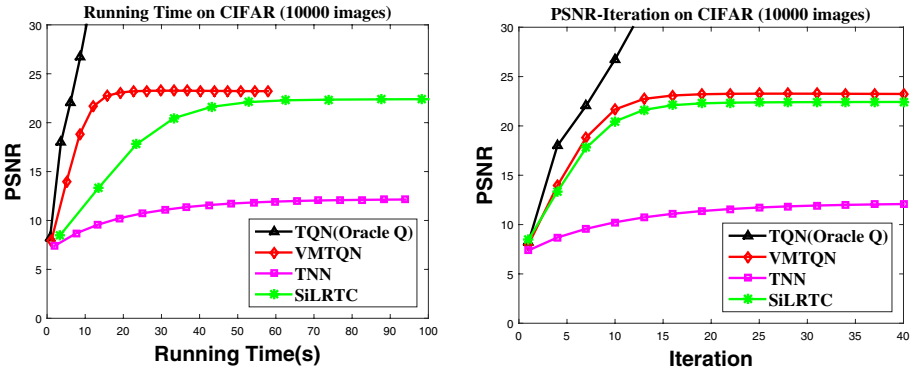

Fig. 4 Running time comparisons of different methods, where $\mathcal{Y} \in \mathbb{R}^{32\times32\times10,000}$ and sampling rate $p = 0.3$

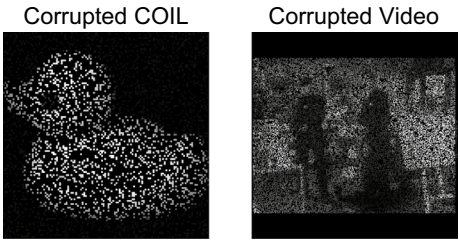

Fig. 5 Examples of the corrupted data in our completion tasks. The left figure is from COIL dataset while the right figure is from the short video. The sampling rate is $p = 0.2$ in the left and $p = 0.5$ in the right

The above analyses confirm that our proposed regularization are data-dependent and can lead to a better low rank structure which makes recover easily.

5.2.3 Running time on CIFAR

As shown in Fig. 4, we test the running times of different models. The two figures indicate that, when $n_3 \gg n_1 n_2$, our VMTQN model has higher computational efficiency in each iteration and better accuracy than TNN and SiLRTC. As mentioned in our previous complexity analysis, VMTQN method has a great speed advantage in this case. Moreover, for the case $n_3 < n_1 n_2$, Fig. 8 implies that setting $r < n_1 n_2$ can balance computational efficiency and recovery accuracy.

5.2.4 COIL-20 and short video from HMDB51

COIL-20 (Nene et al. 1996) contains 1440 images of 20 objects which are taken from different angles. The size of each image is processed as 128×128, which means $\mathcal{Y} \in \mathbb{R}^{128\times128\times1440}$. The upper part of Table 2 shows the results of the numerical experiments. We select a background-changing video from HMDB51 (Kuehne et al. 2011) for the video inpainting task, where $\mathcal{Y} \in \mathbb{R}^{240\times320\times146}$. Figure 2 shows some frames of this video. The lower part of Table 2 shows the results. And Figs. 5, 6 and 7 are the the experimental results of COIL-20 and Short Video from HMDB51, respectively.

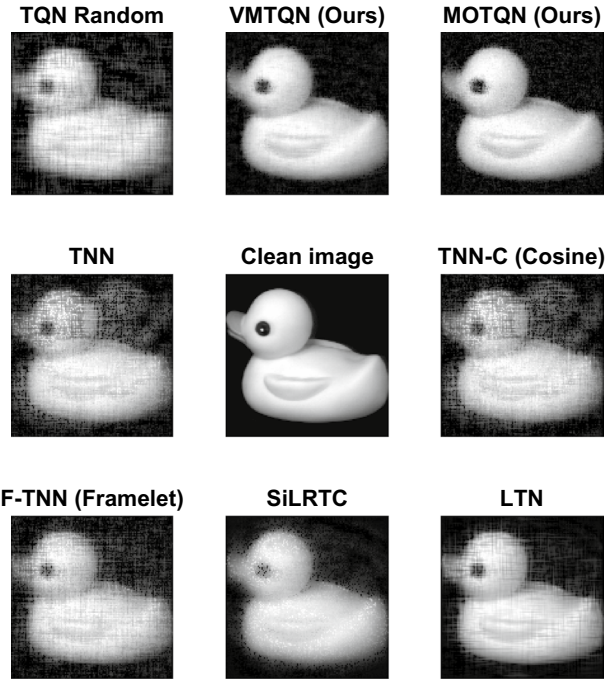

Fig. 6 Examples of COIL completion results. Method names correspond to the top of each figure. The sampling rate $p = 0.2$

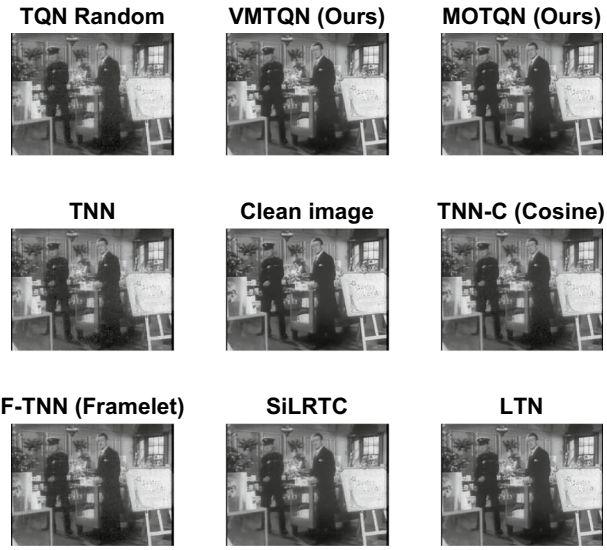

Fig. 7 Examples of video inpainting task with sampling rate $p = 0.5$

From the two visual figures we can see that, our VMTQN method and MOTQN method perform the best among all comparative methods. Especially when the sampling rate $p = 0.2$ in Fig. 6, our methods has significant superiority in visual evaluation. What's more, "Latent Trace Norm" based method performs much better than TNN in COIL, which

Table 2 Comparisons of PSNR results on COIL images and video inpainting with different sampling rates

Sampling rate p	0.1	0.2	0.3	0.4	0.5	0.6
TQN with random $\mathbf{Q}$	16.05	20.07	23.02	25.57	27.95	30.34
TQN with Oracle $\mathbf{Q}$ (ideal)	22.97	25.32	27.18	28.90	30.68	32.51
VMTQN (ours)	**22.79**	**25.34**	**27.29**	**29.08**	**30.86**	**32.74**
MOTQN (ours)	**21.91**	**25.41**	**27.86**	**30.13**	**31.79**	**33.64**
TNN (Fourier) Lu et al. (2018)	19.20	22.08	24.45	26.61	28.72	30.91
TNN-C (cosine) Xu et al. (2019)	19.02	22.11	24.23	27.04	28.95	30.97
TTNN (wavelet) Song et al. (2019)	18.15	21.42	24.47	26.93	29.11	31.10
F-TNN (framelet) Jiang et al. (2020)	17.62	20.58	22.87	24.67	27.41	29.90
Tmac Xu et al. (2017)	19.04	22.48	24.97	26.70	27.91	28.86
SiLRTC Liu et al. (2013)	18.87	21.80	23.89	25.67	27.37	29.14
Latent trace norm Tomioka and Suzuki (2013)	19.09	22.98	25.75	28.11	30.40	32.42
Sampling rate p	0.1	0.2	0.3	0.4	0.5	0.6
TQN with random $\mathbf{Q}$	18.85	22.76	25.87	28.73	31.55	34.48
TQN with Oracle $\mathbf{Q}$ (ideal)	23.44	27.61	31.37	35.11	38.92	42.74
VMTQN (ours)	**23.97**	**28.09**	**31.76**	**35.33**	**39.06**	**42.87**
MOTQN (ours)	**24.10**	**27.88**	**32.24**	**35.19**	**39.28**	**42.65**
TNN (Fourier) Lu et al. (2018)	22.40	25.58	28.28	30.88	33.55	36.41
TNN-C (cosine) Xu et al. (2019)	22.15	25.34	28.17	30.96	33.51	36.62
TTNN (wavelet) Song et al. (2019)	19.80	21.95	24.92	30.13	32.78	36.84
F-TNN (framelet) Jiang et al. (2020)	19.01	23.44	25.94	29.32	32.06	35.13
Tmac Xu et al. (2017)	18.54	22.79	26.08	29.70	31.17	34.26
SiLRTC Liu et al. (2013)	18.42	22.33	25.76	29.15	32.59	36.15
Latent trace norm Tomioka and Suzuki (2013)	18.94	22.72	25.65	28.26	30.79	33.48

Up: the COIL dataset with $\mathcal{Y} \in \mathbb{R}^{128\times128\times1440}$. Down: a short video from HMDB51 with $\mathcal{Y} \in \mathbb{R}^{240\times320\times126}$

validates our assumption that with the help of data-dependent $\mathbf{V}$ tensor trace norm is much more robust than TNN in processing non-smooth data.

Overall, both our methods and t-SVD based methods (e.g., TNN) perform better than the others (e.g., SiLRTC) on these two datasets. It is mainly because the definitions of tensor singular value in tSVD based methods can make better use of the tensor internal structure, and this is also the main difference between tensor Q-nuclear norm (TQN) and sum of the nuclear norm (SNN).

Meanwhile, our method is obviously better than the others at all sampling rates, which reflects the superiority of our data dependent $\mathbf{Q}$.

5.2.5 Influence of r in $\mathbf{Q} \in \mathbb{R}^{n_3\times r}$

Remarks 2 and 3 imply that r of $\mathbf{Q} \in \mathbb{R}^{n_3\times r}$ in VMTQN denotes the apriori assumption of the subspace dimension of the ground-truth. It means that the dimensions of the frontal slice subspace of the true tensor $\mathcal{T}$ (also as the column subspace of mode-3 unfolding matrix $\mathbf{T}_{(3)}$) are no more than r.

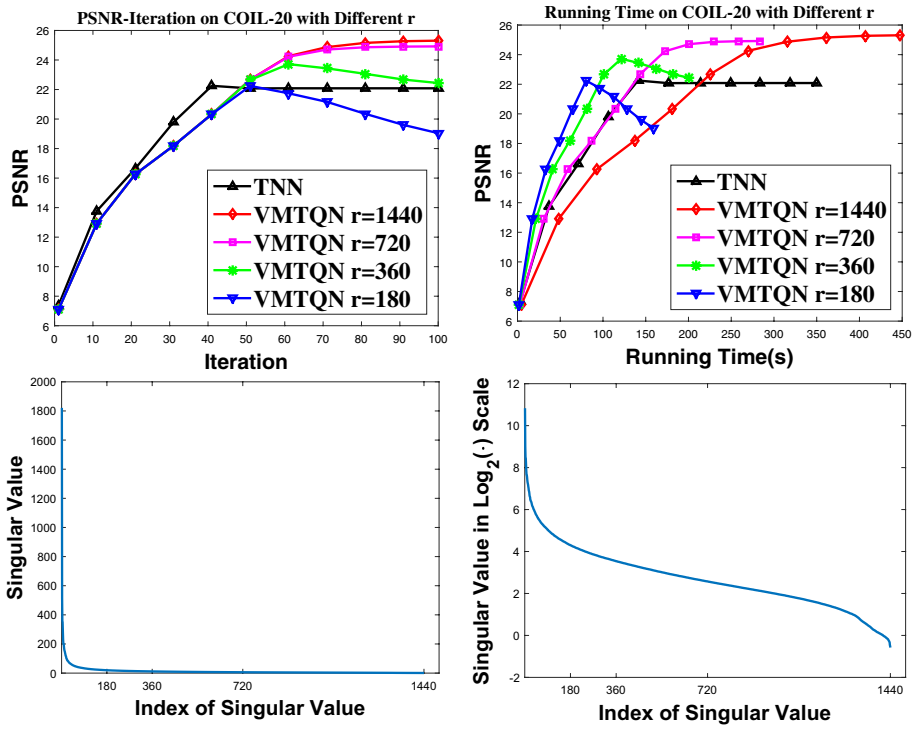

Fig. 8 The relations among running times, different r, and the singular values of $\mathbf{T}_{(3)}$ on COIL, where $p = 0.2$

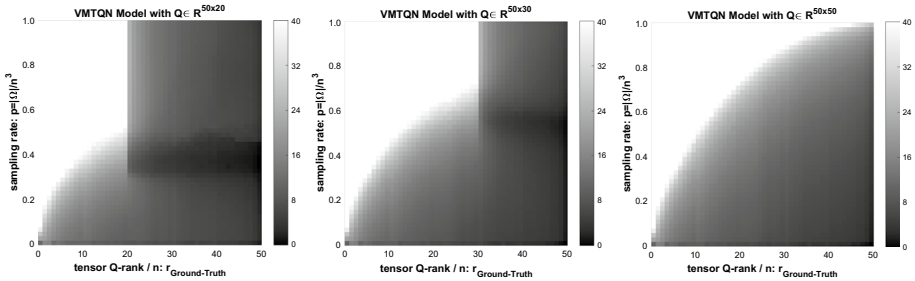

Fig. 9 The gray scale reflects the quality (PSNR) of completion results, where $n_1 = n_2 = n_3 = 50$ and the white area represents a maximum PSNR of 40. There are three different sizes of $\mathbf{Q}$ in VMTQN model to show the influences

Figure 8 illustrates the relations among running times, different r, and the singular values of $\mathbf{T}_{(3)}$. We project the solution $\mathcal{X}_k$ (in Eq. (45)) onto the subspace of $\mathbf{Q}_k$, which means $\hat{\mathcal{X}}_k := \mathcal{X}_k \times_3 (\mathbf{Q}_k \mathbf{Q}_k^\top)$. Meanwhile, under different r in $\mathbf{Q} \in \mathbb{R}^{n_3 \times r}$, Fig. 9 shows the PSNR results of the completion task with varying tensor Q-rank of tensor and varying sampling rate. The settings in Fig. 9 are consistent with those in Sect. 5.1, and only the size of $\mathbf{Q}$ is different.

Fig. 10 Comparisons of PSNR and visualization results of a smooth video inpainting. Up: PSNR results with different sampling rates. Down: visualization results with the sampling rate $p = 0.5$

As shown in the conduct of Fig. 8, the column subspace of $\mathbf{T}_{(3)}$ is more than 360. If $r \leq 360$, the algorithm will converge to a bad point which only has an r-dimensional subspace. Therefore, in our previous experiments, we usually set $r = \min\{n_1 n_2, n_3\}$ to make sure that r is greater than the true tensor's subspace dimension. This apriori assumption is commonly used in factorization-based algorithms. What's more, the running time decreases with the decrease of r. Although $r = 1440$ needs more time to converge than TNN, it obtains a better recovery. And a smaller r does speed up the calculation but harms the accuracy.

The results of Fig. 9 intuitively reflect the selection criterion of r in VMTQN, that is, r should be larger than the subspace dimension of the true tensor to get the exact recovery. According to the constraint $\mathbf{XQQ}^{\top} = \mathbf{X}$ in Sect. 3.1, if the subspace dimension of the true tensor is larger than r, then this constraint can never be satisfied. Additionally, there must be a distance between the output of Algorithm 2 and the truth tensor, which corresponding to the black areas in the upper right corner of the first two sub-figures. From the left two sub-figures we can see that, if the dimension of true tensor is not greater than r, the recovery performance is consistent with that in the third sub-figure. Combined with the above analyses, $r = \min\{n_1 n_2, n_3\}$ can not only save computational efficiency in some cases, but also make the recovery performance of the model in "the white area", corresponding to the exact recovery.

5.3 Smooth data experiments

To verify the effectiveness of our proposed methods in smooth data, we select a video from HMDB51 to conduct the experiments, while the background of this video remains unchanged. Figure 10 shows the PSNR and visualization results of the video inpainting tasks. Here we only compare TNN based method (Lu et al. 2018), since in recent years TNN is considered as a benchmark for handling such smooth data. The results in Fig. 10 show that VMTQN method performs best, and with the increase of sampling rate p, MOTQN method outperforms TNN based method, which means our proposed methods are still competitive in processing smooth data.

6 Conclusions

We analyze the advantages and limitations of the current mainstream low rank regularizers, and then introduce a new definition of data dependent tensor rank named tensor Q-rank. To get a more significant low rank structure w.r.t. rank_Q, we further introduce two explainable selection methods of $\mathbf{Q}$ and make $\mathbf{Q}$ to be a learnable variable w.r.t. the data. Specifically,

maximizing the variance of singular value distribution leads to VMTQN, while minimizing the value of nuclear norm through manifold optimization leads to MOTQN. We provide an envelope of our rank function and apply it to the tensor completion problem. By analyzing the proof of exact recovery theorem,we explain why our method may perform better than TNN based methods in non-smooth data (along the third dimension) with low sampling rates, and conduct experiments to verify our conclusions.

Appendix A: Proof of Lemma 1

Proof Suppose that $\bar{a} = \frac{1}{n}\sum_{i=1}^{n} a_i$, hence the variance of $\{a_1, \ldots, a_n\}$ can be expressed as $\mathrm{Var}[a_i] = \sum_{i=1}^{n}(a_i - \bar{a})^2$. With $\sum_{i=1}^{n} a_i^2 = C$ holds, we have the following:

$$\begin{aligned}\max\ \mathrm{Var}[a_i] &\Rightarrow \max \& \sum_{i=1}^{n}(a_i - \bar{a})^2 \Rightarrow \max\ \sum_{i=1}^{n}(a_i^2 + \bar{a}^2 - 2a_i\bar{a}) \\ &\Rightarrow \max \& \left(\sum_{i=1}^{n} a_i^2\right) + \left(\sum_{i=1}^{n}\bar{a}^2\right) - 2\left(\sum_{i=1}^{n} a_i\bar{a}\right) \\ \Rightarrow \max \& n\bar{a}^2 - 2\bar{a}(n\bar{a}) &\Rightarrow \max\ -n\bar{a}^2 \Rightarrow \min\ \bar{a} \quad (\text{due to } a_i \geq 0).\end{aligned}$$

Moreover, the feasible region of $\{a_1, \ldots, a_n\}$ is an first quadrant Euclidean spherical surface: $\{(a_1, \ldots, a_n) | \sum_{i=1}^{n} a_i^2 = C, a_i \geq 0\}$. Thus the objective function $\bar{a} = \frac{1}{n}\sum_{i=1}^{n} a_i$ is actually a linear hyperplane optimization problem, whose optimal solution contains all intersection of the sphere and each axis, which corresponds to only one non-zero coordinate in $\{a_1, \ldots, a_n\}$. □

Appendix B: Proof of Lemma 2

Proof Firstly, $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ denotes the full Singular Value Decomposition of matrix $\mathbf{X}$ with $\mathbf{U} \in \mathbb{R}^{n_1 \times n_1}$, $\mathbf{\Sigma} \in \mathbb{R}^{n_1 \times n_2}$, and $\mathbf{V} \in \mathbb{R}^{n_2 \times n_2}$. And $\mathbf{P} = \mathbf{V}^\top\mathbf{Q}$ is also an orthogonal matrix, where $\mathbf{P} \in \mathbb{R}^{n_2 \times n_2}$. We use P_{ij} to represent the (i, j)th element of matrix $\mathbf{P}$, and use $\mathbf{p}_i$ to represent the ith column of matrix $\mathbf{P}$. Then $\mathbf{X}\mathbf{Q} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top\mathbf{Q} = \mathbf{U}\mathbf{\Sigma}\mathbf{P}$ holds and we have the following:

$$\|\mathbf{X}\mathbf{Q}\|_{2,1} = \|\mathbf{U}\mathbf{\Sigma}\mathbf{P}\|_{2,1} = \sum_{i=1}^{n_2}\|\mathbf{U}\mathbf{\Sigma}\mathbf{p}_i\|_2 = \sum_{i=1}^{n_2}\|\mathbf{\Sigma}\mathbf{p}_i\|_2. \tag{56}$$

If $n_1 \geq n_2$, let $\sigma_i = \mathbf{\Sigma}_{ii}$ be the (i, i)th element value of $\mathbf{\Sigma}$ with $i = 1, \ldots, n_2$. Or if $n_1 < n_2$, let $\mathbf{\Sigma}' = \begin{pmatrix}\mathbf{\Sigma} \\ \mathbf{0}\end{pmatrix} \in \mathbb{R}^{n_2 \times n_2}$ and $\sigma_i = \mathbf{\Sigma}'_{ii}$ with $i = 1, \ldots, n_2$. In this case, $\sum_{i=1}^{n_2}\|\mathbf{\Sigma}\mathbf{p}_i\|_2 = \sum_{i=1}^{n_2}\|\mathbf{\Sigma}'\mathbf{p}_i\|_2$. Thus, we can always get $\{\sigma_1, \ldots, \sigma_{n_2}\}$ and have the equation $\sum_{i=1}^{n_2}\|\mathbf{\Sigma}\mathbf{p}_i\|_2 = \sum_{i=1}^{n_2}\sqrt{\sum_{j=1}^{n_2}(\sigma_j P_{ji})^2}$.

We then prove that $\mathbf{P} = \mathbf{I}$ optimize the problem (16). By using Eq. (56), the objective function can be written as $\sum_{i=1}^{n_2}\|\mathbf{\Sigma}\mathbf{p}_i\|_2$. We give the following deduction:

$$\sum_{i=1}^{n_2}\|\mathbf{\Sigma p}_i\|_2=\sum_{i=1}^{n_2}\sqrt{\sum_{j=1}^{n_2}(\sigma_j P_{ji})^2}\overset{(a)}{=}\sum_{i=1}^{n_2}\sqrt{\sum_{j=1}^{n_2}(\sigma_j P_{ji})^2\times\sum_{j=1}^{n_2}P_{ji}^2}$$
$$\overset{(b)}{\geq}\sum_{i=1}^{n_2}\sum_{j=1}^{n_2}(\sigma_j P_{ji}^2)\overset{(c)}{=}\sum_{j=1}^{n_2}\sigma_j\left(\sum_{i=1}^{n_2}P_{ji}^2\right)\overset{(d)}{=}\sum_{j=1}^{n_2}\sigma_j.$$

(a) holds due to that $\mathbf{P}$ is an orthogonal matrix with normalized columns. (b) holds because of Cauchy inequality. (c) holds with exchanging the order of two summations. Finally (d) holds owing to the row normalization of $\mathbf{P}$. Notice that the equality in (b) holds if and only if the two vectors $(\sigma_1 P_{1i},\ldots,\sigma_{n_2}P_{n_2 i})$ and $(P_{1i},\ldots,P_{n_2 i})$ are parallel. It can be seen that when $\mathbf{P}=\mathbf{I}$, the condition are satisfied. In other words, $\mathbf{V}^\top\mathbf{Q}=\mathbf{I}$ optimize the problem (16), which implies $\mathbf{Q}=\mathbf{V}$. □

Appendix C: Proof of Theorem 2

Proof We divide $r=\min\{n_1,n_2\}$ into two cases and prove them respectively. And we use the same notation as in the previous proofs.

(1) If $n_1<n_2$ and $r=n_1$, then $\mathbf{U}\in\mathbb{R}^{n_1\times n_1}$, $\mathbf{V}\in\mathbb{R}^{n_2\times n_1}$, and $\mathbf{Q}\in\mathbb{R}^{n_2\times n_1}$. In this case, $\mathbf{\Sigma}\in\mathbb{R}^{n_1\times n_1}$. Let $\mathbf{\Sigma}'=\begin{pmatrix}\Sigma & \mathbf{0}\end{pmatrix}\in\mathbb{R}^{n_1\times n_2}$, $\mathbf{V}'=\begin{pmatrix}\mathbf{V} & \mathbf{V}_\perp\end{pmatrix}\in\mathbb{R}^{n_2\times n_2}$, and $\mathbf{Q}'=\begin{pmatrix}\mathbf{Q} & \mathbf{Q}_\perp\end{pmatrix}\in\mathbb{R}^{n_2\times n_2}$. Note that the constraint $\mathbf{XQQ}^\top=\mathbf{X}$ in Eq. (17) implies $\mathbf{V}^\top\mathbf{Q}_\perp=\mathbf{0}$ and $\mathbf{V}_\perp^\top\mathbf{Q}=\mathbf{0}$, then we have the following:

$$\|\mathbf{XQ}\|_{2,1}=\|\mathbf{U\Sigma V}^\top\mathbf{Q}\|_{2,1}=\|\mathbf{\Sigma V}^\top\mathbf{Q}\|_{2,1}=\|\mathbf{\Sigma}'\mathbf{V}'^\top\mathbf{Q}'\|_{2,1}. \tag{57}$$

That is to say, minimize $\|\mathbf{XQ}\|_{2,1}$ w.r.t. $\mathbf{Q}$ in Eq. (17) is equivalent to minimize $\|\mathbf{\Sigma}'\mathbf{V}'^\top\mathbf{Q}'\|_{2,1}$ w.r.t. $\mathbf{Q}'$ under the constraints $\mathbf{V}^\top\mathbf{Q}_\perp=\mathbf{0}$ and $\mathbf{V}_\perp^\top\mathbf{Q}=\mathbf{0}$. By using Lemma 2, $\mathbf{Q}'=\mathbf{V}'$ minimize the objective function $\|\mathbf{\Sigma}'\mathbf{V}'^\top\mathbf{Q}'\|_{2,1}$, which also satisfies the constraints. In other words, $\mathbf{Q}=\mathbf{V}$ optimize the problem 17.

(2) If $n_1\geq n_2$ and $r=n_2$, then $\mathbf{U}\in\mathbb{R}^{n_1\times n_2}$, $\mathbf{V}\in\mathbb{R}^{n_2\times n_2}$, and $\mathbf{Q}\in\mathbb{R}^{n_2\times n_2}$. In this case, we have

$$\|\mathbf{XQ}\|_{2,1}=\|\mathbf{U\Sigma P}\|_{2,1}=\sum_{i=1}^{n_2}\|\mathbf{U\Sigma p}_i\|_2=\sum_{i=1}^{n_2}\|\mathbf{\Sigma p}_i\|_2.$$

The remaining proofs are similar to the details in "Appendix B". □

Appendix D: Proof of Lemma 3

Proof Let $g(\tau)=f(\mathbf{Q}(\tau))=\|\mathcal{X}\|_{Q(\tau),*}$ and $\mathbf{Q}(\tau)\approx\left(\mathbf{I}-\tau\mathbf{A}+\frac{\tau^2}{2}\mathbf{A}^2\right)\mathbf{Q}_k$, where $\mathbf{A}$ is defined in Eq. (32). We consider the following approximation:

$$g(\tau) = f(\mathbf{Q}(\tau)) \approx g(0) + \left\langle \frac{\partial f(\mathbf{Q}(\tau))}{\partial \mathbf{Q}(\tau)}\Big|_{\tau=0}, \mathbf{Q}(\tau) - \mathbf{Q}(0) \right\rangle = g(0) + \left\langle \mathbf{X}_{(3)}^{\top}\mathbf{H}_{(3)}, \mathbf{Q}(\tau) - \mathbf{Q}(0) \right\rangle, \tag{58}$$

where $\mathbf{Q}(0) = \mathbf{Q}_k$ and then Eq. (31) ensure $\frac{\partial f(\mathbf{Q}(\tau))}{\partial \mathbf{Q}(\tau)}\Big|_{\tau=0} = \mathbf{X}_{(3)}^{\top}\mathbf{H}_{(3)}$. Then we have:

$$g(\tau) \approx \left\langle \mathbf{X}_{(3)}^{\top}\mathbf{H}_{(3)}, \left(\mathbf{I} - \tau\mathbf{A} + \frac{\tau^2}{2}\mathbf{A}^2\right)\mathbf{Q}_k \right\rangle + C_\tau, \tag{59}$$

where C_τ is a constant independent of τ. Then the first and the second order derivatives of $g(\tau)$ evaluated at 0 can be estimated as follows:

$$g'(0) \approx \left\langle \mathbf{X}_{(3)}^{\top}\mathbf{H}_{(3)}, -\mathbf{A}\mathbf{Q}_k \right\rangle, \text{ and } g''(0) \approx \left\langle \mathbf{X}_{(3)}^{\top}\mathbf{H}_{(3)}, \mathbf{A}^2\mathbf{Q}_k \right\rangle, \tag{60}$$

□

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10994-021-05987-8.

Acknowledgements Funding was provided by Key-Area Research and Development Program of Guangdong Province (No. 2019B121204008), NSF China (grant nos. 61625301 and 61731018), Major Scientific Research Project of Zhejiang Lab (grant nos. 2019KB0AC01 and 2019KB0AB02), Beijing Academy of Artificial Intelligence, and Qualcomm.

References

Absil, P.-A., Mahony, R., & Sepulchre, R. (2009). *Optimization algorithms on matrix manifolds*. Princeton, NJ: Princeton University Press.

Cai, J.-F., Chan, R. H., & Shen, Z. (2008). A framelet-based image inpainting algorithm. *Applied and Computational Harmonic Analysis, 24*(2), 131–149.

Candès, E. J., & Recht, B. (2009). Exact matrix completion via convex optimization. *Foundations of Computational Mathematics, 9*(6), 717.

Candès, E. J., & Tao, T. (2010). The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory, 56*(5), 2053–2080.

Edelman, A., Arias, T. A., & Smith, S. T. (1998). The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications, 20*(2), 303–353.

Friedland, S., & Lim, L.-H. (2018). Nuclear norm of higher-order tensors. *Mathematics of Computation, 87*(311), 1255–1281.

Fu, Y., Gao, J., Tien, D., Lin, Z., & Hong, X. (2016). Tensor lrr and sparse coding-based subspace clustering. *IEEE Transactions on Neural Networks and Learning Systems, 27*(10), 2120–2133.

Håstad, J. (1990). Tensor rank is NP-complete. *Journal of Algorithms, 11*(4), 644–654.

Hillar, C. J., & Lim, L.-H. (2013). Most tensor problems are NP-hard. *Journal of the ACM, 60*(6), 45.

Hitchcock, F. L. (1927). The expression of a tensor or a polyadic as a sum of products. *Studies in Applied Mathematics, 6*(1–4), 164–189.

Hitchcock, F. L. (1928). Multiple invariants and generalized rank of a p-way matrix or tensor. *Journal of Mathematics and Physics, 7*(1–4), 39–79.

Hu, W., Tao, D., Zhang, W., Xie, Y., & Yang, Y. (2016). The twist tensor nuclear norm for video completion. *IEEE Transactions on Neural Networks and Learning Systems, 28*(12), 2961–2973.

Jiang, T.-X., Huang, T.-Z., Zhao, X.-L., Ji, T.-Y., & Deng, L.-J. (2018). Matrix factorization for low-rank tensor completion using framelet prior. *Information Sciences, 436,* 403–417.

Jiang, T.-X., Ng, M. K., Zhao, X.-L., & Huang, T.-Z. (2020). Framelet representation of tensor nuclear norm for third-order tensor completion. *IEEE Transactions on Image Processing, 29,* 7233–7244.
Kasai, H., & Mishra, B. (2016). Low-rank tensor completion: A Riemannian manifold preconditioning approach. In *International conference on machine learning,* pp. 1012–1021.
Kernfeld, E., Kilmer, M., & Aeron, S. (2015). Tensor-tensor products with invertible linear transforms. *Linear Algebra and its Applications, 485,* 545–570.
Kiers, H. A. (2000). Towards a standardized notation and terminology in multiway analysis. *Journal of Chemometrics, 14*(3), 105–122.
Kilmer, M. E., Braman, K., Hao, N., & Hoover, R. C. (2013). 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.
Kilmer, M. E., & Martin, C. D. (2011). Factorization strategies for third-order tensors. *Linear Algebra and its Applications, 435*(3), 641–658.
Kolda, T. G., & Bader, B. W. (2009). Tensor decompositions and applications. *SIAM Review, 51*(3), 455–500.
Kong, H., Xie, X., & Lin, Z. (2018). t-Schatten-p norm for low-rank tensor recovery. *IEEE Journal of Selected Topics in Signal Processing, 12*(6), 1405–1419.
Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images, tech. rep., Citeseer.
Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., & Serre, T. (2011). HMDB: A large video database for human motion recognition. In *IEEE international conference on computer vision,* pp. 2556–2563.
Li, C., Guo, L., Tao, Y., Wang, J., Qi, L., & Dou, Z. (2016). Yet another Schatten norm for tensor recovery. In *International conference on neural information processing,* pp. 51–60.
Lin, Z., Liu, R., & Su, Z. (2011). Linearized alternating direction method with adaptive penalty for low-rank representation. *Advances in Neural Information Processing Systems,* 612–620.
Lin, Z., Liu, R., & Li, H. (2015). Linearized alternating direction method with parallel splitting and adaptive penalty for separable convex programs in machine learning. *Machine Learning, 99*(2), 287.
Liu, J., Musialski, P., Wonka, P., & Ye, J. (2013). Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*(1), 208–220.
Liu, Y., Shang, F., Fan, W., Cheng, J., & Cheng, H. (2015). Generalized higher order orthogonal iteration for tensor learning and decomposition. *IEEE Transactions on Neural Networks and Learning Systems, 27*(12), 2551–2563.
Lu, C., Feng, J., Chen, Y., Liu, W., Lin, Z., & Yan, S. (2016). Tensor robust principal component analysis: Exact recovery of corrupted low-rank tensors via convex optimization. In *Proceedings of the IEEE conference on computer vision and pattern recognition,* pp. 5249–5257.
Lu, C., Feng, J., Lin, Z., & Yan, S. (2018). Exact low tubal rank tensor recovery from gaussian measurements. In *International conference on artificial intelligence*.
Lu, C., Peng, X., & Wei, Y. (2019). Low-rank tensor completion with a new tensor nuclear norm induced by invertible linear transforms. In *Proceedings of the IEEE conference on computer vision and pattern recognition,* pp. 5996–6004.
Lu, C., Feng, J., Chen, Y., Liu, W., Lin, Z., & Yan, S. (2019). Tensor robust principal component analysis with a new tensor nuclear norm. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 42*(4), 925–938.
Lu, C., Feng, J., Yan, S., & Lin, Z. (2017). A unified alternating direction method of multipliers by majorization minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 40*(3), 527–541.
Nene, S. A., Nayar, S. K., Murase, H., *et al.* (1996). Columbia object image library (coil-20).
Ng, M. K., Chan, R. H., & Tang, W.-C. (1999). A fast algorithm for deblurring models with Neumann boundary conditions. *SIAM Journal on Scientific Computing, 21*(3), 851–866.
Petersen, K. B., Pedersen, M. S., et al. (2008). The matrix cookbook. *Technical University of Denmark, 7*(15), 510.
Romera-Paredes, B., & Pontil, M. (2013). A new convex relaxation for tensor completion. *Advances in Neural Information Processing Systems,* 2967–2975.
Song, G., Ng, M. K., & Zhang, X. (2019). Robust tensor completion using transformed tensor svd. arXiv preprint arXiv:1907.01113.
Tomioka, R., & Suzuki, T. (2013). Convex tensor decomposition via structured Schatten norm regularization. In *Advances in neural information processing systems*, pp. 1331–1339.
Tomioka, R., Hayashi, K., & Kashima, H. (2010). On the extension of trace norm to tensors. In *NIPS workshop on tensors, kernels, and machine learning*, p. 7.

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

Wen, Z., & Yin, W. (2013). A feasible method for optimization with orthogonality constraints. *Mathematical Programming, 142*(1–2), 397–434.

Wimalawarne, K., Sugiyama, M., & Tomioka, R. (2014). Multitask learning meets tensor factorization: Task imputation via convex optimization. *Advances in Neural Information Processing Systems,* 2825–2833.

Xu, W.-H., Zhao, X.-L., & Ng, M. (2019). A fast algorithm for cosine transform based tensor singular value decomposition. arXiv preprint arXiv:1902.03070.

Xu, Y., Hao, R., Yin, W., & Su, Z. (2017). Parallel matrix factorization for low-rank tensor completion. *Inverse Problems & Imaging, 9*(2), 601–624.

Xu, Y., & Yin, W. (2015). A block coordinate descent method for regularized multi-convex optimization with applications to nonnegative tensor factorization and completion. *SIAM Journal on Imaging Sciences, 6*(3), 1758–1789.

Yin, M., Gao, J., Xie, S., & Guo, Y. (2018). Multiview subspace clustering via tensorial t-product representation. *IEEE Transactions on Neural Networks and Learning Systems, 30*(3), 851–864.

Yuan, M., & Zhang, C.-H. (2016). On tensor completion via nuclear norm minimization. *Foundations of Computational Mathematics, 16*(4), 1031–1068.

Zhang, Z., Ely, G., Aeron, S., Hao, N., & Kilmer, M. (2014). Novel methods for multilinear data completion and de-noising based on tensor-SVD. In *Proceedings of the IEEE conference on computer vision and pattern recognition,* pp. 3842–3849.

Zhang, Z., & Aeron, S. (2017). Exact tensor completion using t-SVD. *IEEE Transactions on Signal Processing, 65*(6), 1511–1526.

Zhou, P., Lu, C., Lin, Z., & Zhang, C. (2018). Tensor factorization for low-rank tensor completion. *IEEE Transactions on Image Processing, 27*(3), 1152–1163.