Score-Based Model for Low-Rank Tensor Recovery

Zhengyun Cheng[1], Changhao Wang[1], Guanwen Zhang[1*], Yi Xu[2], Wei Zhou[1*], Xiangyang Ji[3]

[1]School of Electronics and Information, Northwestern Polytechnical University
[2]School of Control Science and Engineering, Dalian University of Technology
[3]Department of Automation, Tsinghua University
guanwen.zh@nwpu.edu.cn, zhouwei@nwpu.edu.cn

Abstract

Low-rank tensor decompositions (TDs) provide an effective framework for multiway data analysis. Traditional TD methods rely on predefined structural assumptions, such as CP or Tucker decompositions. From a probabilistic perspective, these methods effectively model the relationships between latent factors and the low-rank tensor using Dirac delta distributions. However, tensor low-rank decomposition is inherently non-unique, leading to a multimodal distribution over possible solutions. Critically, such prior knowledge is rarely available in practical scenarios, particularly regarding the optimal rank structure and contraction rules. To address this issue, we propose a score-based model that eliminates the need for predefined structural or distributional assumptions, enabling the learning of compatibility between tensors and latent factors. Specifically, a neural network is designed to learn the energy function, which is optimized via score matching to capture the gradient of the joint log-probability of tensor entries and latent factors. Our method allows for modeling structures and distributions beyond the Dirac delta assumption. Moreover, integrating the block coordinate descent (BCD) algorithm with the proposed smooth regularization enables the model to perform both tensor completion and denoising. Experimental results demonstrate significant performance improvements across various tensor types, including sparse and continuous-time tensors, as well as visual data.

Code — https://github.com/CZY-Code/ScoreTR

1 Introduction

Tensor decompositions serve as powerful tools for analyzing high-order and high-dimensional data, aiming to capture the inter-dependencies among different modes by utilizing multiple shared latent factors. TDs have demonstrated remarkable success in various machine learning tasks, including data imputation (Zhe et al. 2016; Fang et al. 2021), factor analysis (Chen, Yang, and Zhang 2022), time-series forecasting (Miller, Rabusseau, and Terilla 2021), model compression (Novikov et al. 2015; Dai et al. 2023), generative models (Glasser et al. 2019; Kuznetsov et al. 2019) among others.

*Corresponding author

Mathematically, given an incomplete or noisy observation tensor $\hat{\mathcal{X}}$, the TDs model aims to recover unknown tensor $\mathcal{X}$ and noise component $\mathcal{S}$. Using Bayesian rule, the TDs model can be formulated as

$$p(\mathcal{X}, \mathcal{Z}, \mathcal{S}|\hat{\mathcal{X}}) \propto p(\hat{\mathcal{X}}|\mathcal{X}, \mathcal{Z}, \mathcal{S})p(\mathcal{S})p(\mathcal{X}|\mathcal{Z})p(\mathcal{Z}). \quad (1)$$

The above equation holds because the noise $\mathcal{S}$ and the data $\mathcal{X}$ are independent. Existing TD methods typically specify prior $p(\mathcal{Z})$ on latent factors $\mathcal{Z}$, and incorporate predefined contraction rules $\mathrm{CR}(\mathcal{Z})$ and noise prior $p(\mathcal{S})$ to maximize the posterior probability of the observed tensor. This can be formulated as minimizing the negative log-posterior of Eq. (1)

$$\begin{aligned} & -\log p(\mathcal{X}, \mathcal{Z}, \mathcal{S}|\hat{\mathcal{X}}) \\ = & -\log p(\hat{\mathcal{X}}|\mathcal{X}, \mathcal{Z}, \mathcal{S}) && \Rightarrow \|\hat{\mathcal{X}} - \mathcal{X} - \mathcal{S}\|_F && \text{(likelihood)} \\ & -\log p(\mathcal{S}) && \Rightarrow \|\mathcal{S}\|_{\ell_1} && \text{(noise prior)} \\ & -\log p(\mathcal{X}|\mathcal{Z}) && \Rightarrow \|\mathcal{X} - \mathrm{CR}(\mathcal{Z})\|_F && \text{(Dirac } \delta\text{)} \\ & -\log p(\mathcal{Z}). && \Rightarrow \mathfrak{R}(\mathcal{Z}) && \text{(factor prior)} \end{aligned} \quad (2)$$

The second term models sparse or Laplacian noise, often regularized using the ℓ_1-norm $\|\hat{\mathcal{X}} - \mathcal{X}\|_{\ell_1}$. The third term represents the contraction rule that defines the generative process from latent factors to the tensor, which is typically modeled as a Dirac delta distribution $p(\mathcal{X}|\mathcal{Z}) = \delta(\mathcal{X} - \mathrm{CR}(\mathcal{Z}))$, and relaxed to an F-norm penalty $\|\mathcal{X} - \mathrm{CR}(\mathcal{Z})\|_F$. The last term, $\mathfrak{R}(\mathcal{Z})$, serves as a low-rank regularizer.

From the perspective of modeling $p(\mathcal{X}|\mathcal{Z})$, the traditional approaches mainly focus on identifying suitable contraction structures, including classical models such as CANDECOMP/PARAFAC (CP) (Kolda and Bader 2009), Tucker (Zhou et al. 2015), and tensor train (Oseledets 2011), along with their variants (Zhao et al. 2016; Zheng et al. 2021; Wu et al. 2022). However, in real-world applications, such prior knowledge, e.g., the definition of tensor rank and contraction rules, is often unavailable. Moreover, the low-rank prior can vary significantly across different domains. Consequently, selecting an appropriate TD model for a specific dataset can be challenging. Recent work, referred to as tensor network structure search (Li and Sun 2020; Li et al. 2022), has demonstrated that choosing an appropriate contraction rule can substantially improve factorization performance. Another promising direction is learning nonlinear mappings

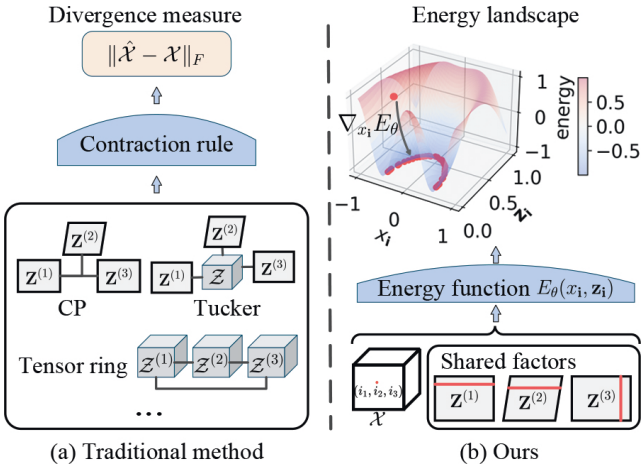
directly from the observed tensor, using techniques such as nonparametric models (Chu and Ghahramani 2009; Xu, Yan, and Qi 2012; Zhe et al. 2016) and deep neural networks (Liu et al. 2019; Fang et al. 2021; Luo et al. 2023; Fan 2021). While nonlinear TDs have succeeded in relaxing rigid structural assumptions, they still implicitly model low-rank decomposition as a Dirac delta distribution conditioned on parametric contraction functions. This formulation is inherently limited, as tensor low-rank decomposition is fundamentally non-unique, leading to a multimodal distribution that goes beyond the Dirac delta assumption.

From the perspective of modeling the latent factor prior $p(\mathcal{Z})$, a Gaussian prior is often imposed on the latent factors, while the observed entries are modeled using a Gaussian likelihood (Rai et al. 2014; Zhao, Zhang, and Cichocki 2015) or a Gaussian process (Xu, Yan, and Qi 2012; Zhe et al. 2016). These priors can also be realized through norm-based regularizations, such as the nuclear norm $\|\mathbf{X}\|_*$, the Schatten-p norm $\|\mathcal{X}\|_{S_p}$ (Giampouras et al. 2020), or the Frobenius norm of the latent factors (Fan et al. 2023). However, in real-world applications, the latent factors may originate from unknown or complex distributions, and observations can be generated through intricate, multi-modal processes. In the absence of knowledge about the true generative mechanism, such restrictive distributional assumptions can introduce model bias, limit the expressiveness of TD models, and ultimately lead to inaccurate estimations.

Recently, (Tao, Tanaka, and Zhao 2023) proposed to model the joint probability $p(\mathcal{X}, \mathcal{Z})$ of the tensor and its latent factors from an energy-based perspective. A variational Gaussian distribution was employed to approximate the true posterior of the latent factors, and a noise-contrastive loss was used for optimization. However, this method heavily relies on high-quality pairwise noise to avoid energy collapse and accurate posterior estimation, and it is not applicable to denoising task.

To address the above issues, this paper proposes to model the gradient of the joint probability distribution via a parameterized energy function, leveraging score matching with multiple noise levels for optimization, as shown in Figure 1. We capture low-rank structure through trainable shared latent factors on each mode, avoiding the limitations imposed by fixed contraction rules. The model is optimized using the Adam algorithm, which eliminates the need for complex iterative solvers such as the Augmented Lagrangian Method or the Alternating Direction Method of Multipliers, commonly employed in traditional tensor decomposition methods. The proposed method offers several key advantages:

- We introduce a flexible framework capable of adapting to diverse joint distributions between tensors and latent factors. Moving beyond the conventional separate modeling of $p(\mathcal{X}|\mathcal{Z})$ and $p(\mathcal{Z})$, thereby removing the constraints imposed by the Dirac delta distribution.
- By modeling the gradient of the joint probability, we avoid the need for explicit posterior approximation of the latent factors and bypass complex optimization procedures. Score matching is introduced to directly learn the gradient of the energy landscape.



(a) Traditional method (b) Ours

Figure 1: (a) The traditional method model the Dirac delta distribution of latent factors and tensors through predefined contraction rules. (b) The proposed method model the gradient of the log joint probability density function with respect to the entry from energy perspective.

- Integrating the model into an BCD framework allows for both tensor completion and denoising. Moreover, we proposed a smooth regularization from energy function perspective for denoising.
- Extensive experiments on synthetic and real-world datasets demonstrate the significantly superior performance of the proposed method.

2 Preliminaries

Notations

We adopt similar notations with (Kolda and Bader 2009). Throughout the paper, we use lowercase letters, bold lowercase letters, bold capital letters and calligraphic capital letters to represent scalars, vectors, matrices and tensors, e.g., $x, \mathbf{x}, \mathbf{X}, \mathcal{X}$. Tensors refer to multi-way arrays which generalize matrices. For a D-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_D}$, we denote its $(i_1, \cdots, i_D)$-th entry as $x_{\mathbf{i}}$.

Denoising Score Matching with Langevin Dynamics

Let $p_\sigma(\tilde{x}|x) := \mathcal{N}(\tilde{x}; x, \sigma^2)$ be a perturbation kernel, and $p_\sigma(\tilde{x}|x) := \int p_d(x) p_\sigma(\tilde{x}|x)\, \mathrm{d}x$, where $p_d(x)$ denotes the data distribution. Consider a sequence of positive noise scales $\sigma_{min} = \sigma_1 < \sigma_2 < \cdots \sigma_L = \sigma_{max}$. Typically, σ_{min} is small enough such that $p_{\sigma_{min}}(x) \approx p_d(x)$, and σ_{max} is large enough such that $p_{\sigma_{max}}(x) \approx \mathcal{N}(x; 0, \sigma^2_{max})$. (Song and Ermon 2019) propose to train a Noise Conditional Score Network, denoted by $\boldsymbol{s_\theta}(x, \sigma)$, with a weighted sum of denoising score matching (Vincent 2011) objectives

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum_{l=1}^{L} \Big\{ \sigma_l^2 \mathbb{E}_{p_{\sigma_l}(\tilde{x}|x) p_d(x)} \Big[\|\nabla_{\tilde{x}} \log p_\theta(\tilde{x}, \sigma_l) - \nabla_{\tilde{x}} \log p_{\sigma_l}(\tilde{x}|x)\|_2^2 \Big] \Big\}. \quad (3)$$

Given sufficient data and model capacity, the optimal score-based model $\nabla_{\tilde{x}} \log p_{\theta^*}(\tilde{x}, \sigma_l)$ matches $\nabla_x \log p_\sigma(x)$ almost everywhere for $\sigma \in \{\sigma_l\}_{l=1}^L$. For sampling, (Song and Ermon 2019) run T steps of Langevin MCMC to get a sample for each $p_{\sigma_l}(x)$ sequentially

$$x_l^k = x_l^{k-1} + \alpha_l \nabla_{\tilde{x}} \log p_{\theta^*}(x_l^{k-1}, \sigma_l) + \sqrt{2\alpha_l}\epsilon_k, \quad (4)$$

where $k = 1, 2, \cdots, K$, $\alpha_l > 0$ is the step size, and ϵ_k is standard normal. The above step is repeated for $l = L, L-1, \cdots, 1$ in turn with $x_N^0 \sim \mathcal{N}(x|0, \sigma_{max})$ and $x_l^0 = x_{l+1}^K$ when $l < L$. As $K \to \infty$ and $\alpha_l \to 0$ for all l, x_1^K becomes an exact sample from $p_{\sigma_{min}}(x) \approx p_d(x)$ under some regularity conditions.

3 Methodology

Given a noisy or incomplete observation tensor $\hat{\mathcal{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_D}$, our method aims to recover the low-rank tensor $\mathcal{X}$ through D latent factor matrices $\mathcal{Z} = \{\mathbf{Z}^1, \ldots, \mathbf{Z}^D\}$, and the noise tensor $\mathcal{S}$ when applicable for denoising tasks. We denote the latent factor associated with the $(i_1, \ldots, i_D)$-th entry as $\mathbf{Z}_\mathbf{i} = [\mathbf{z}_{i_1}^1, \ldots, \mathbf{z}_{i_D}^D] \in \mathbb{R}^{D \times R}$, where $\mathbf{z}_{i_d}^d \in \mathbb{R}^R$ represents the i_d-th row of $\mathbf{Z}^d$. Thus the entries with partially identical coordinates share the some row of the latent factors. We further define $\mathbf{z}_\mathbf{i} \in \mathbb{R}^{DR}$ as the flattened vectorization of $\mathbf{Z}_\mathbf{i}$. Instead of relying on traditional, intractable contraction rules or explicit priors of latent factor to model $p(\mathcal{X}|\mathcal{Z})p(\mathcal{Z})$, we extend score-based methods to learn the first-order gradient of the joint log-probability density function with respect to tensor entry $x_\mathbf{i}$, conditioned on the latent factors determined by its coordinate. Specifically, we model $\nabla_{x_\mathbf{i}} \log p(x_\mathbf{i}, \mathbf{z}_\mathbf{i})$, referred to as the score of the tensor entry located in $\mathbf{i} = (i_1, \ldots, i_D)$. Furthermore, we incorporate a BCD algorithm to enable both tensor completion and denoising applications within a unified framework.

Density Estimation with Denoising Score Matching for Tensor Entry

It is worth noting that the score function from the perspective of tensor entries can be rewritten as: $\nabla_{x_\mathbf{i}} \log p(x_\mathbf{i} \mid \mathbf{z}_\mathbf{i}) = \nabla_{x_\mathbf{i}} \log \frac{p(x_\mathbf{i}, \mathbf{z}_\mathbf{i})}{p(\mathbf{z}_\mathbf{i})} = \nabla_{x_\mathbf{i}} \log p(x_\mathbf{i}, \mathbf{z}_\mathbf{i})$. We model the joint distribution using an energy-based formulation:

$$p(x_\mathbf{i}, \mathbf{z}_\mathbf{i}) = \frac{e^{-E(x_\mathbf{i}, \mathbf{z}_\mathbf{i})}}{\int e^{-E(x_\mathbf{i}, \mathbf{z}_\mathbf{i})} \, \mathrm{d}x_\mathbf{i} \, \mathrm{d}\mathbf{z}_\mathbf{i}}, \quad (5)$$

where $E(x_\mathbf{i}, \mathbf{z}_\mathbf{i})$ denotes the joint energy function, and the denominator is the partition function that ensures the validity of the joint probability density function. Following standard assumptions in tensor decomposition, we further assume independence among all tensor entries, with dependencies captured through shared latent factors. We apply multiple noise levels to align the denoising score with the energy gradient and enhance local curvature, based on denoising score matching. For a given noise scale σ, the denoising score matching objective becomes:

$$\ell(\theta, \mathcal{Z}; \sigma) \triangleq \frac{1}{2} \mathbb{E}_{\tilde{x}_\mathbf{i} \sim \mathcal{N}(x_\mathbf{i}, \sigma^2), \, \mathbf{i} \sim \Omega} \left[\left\| \frac{\tilde{x}_\mathbf{i} - x_\mathbf{i}}{\sigma^2} + \nabla_{\tilde{x}_\mathbf{i}} E_\theta(\tilde{x}_\mathbf{i}, \mathbf{z}_\mathbf{i}) \right\|_F^2 \right], \quad (6)$$

where the $\nabla_{\tilde{x}_\mathbf{i}} E_\theta(\tilde{x}_\mathbf{i}, \mathbf{z}_\mathbf{i}) = -\nabla_{\tilde{x}_\mathbf{i}} \log p_\theta(\tilde{x}_\mathbf{i}, \mathbf{z}_\mathbf{i})$, and Ω denotes the set of observed indices. We combine the objectives in Eq. (6) across all noise scales $\sigma \in \{\sigma_l\}_{l=1}^L$ to form a unified loss:

$$\mathcal{L}_D(\theta, \mathcal{Z}; \{\sigma_l\}_{l=1}^L) \triangleq \frac{1}{L} \sum_{l=1}^L \sigma_l^2 \, \ell(\theta, \mathcal{Z}; \sigma_l). \quad (7)$$

Assuming the energy function $E_\theta(x_\mathbf{i}, \mathbf{z}_\mathbf{i})$ has sufficient capacity, $E_{\theta^*}(x_\mathbf{i}, \mathbf{z}_\mathbf{i})$ minimizes Eq. (7) if and only if $\|x_\mathbf{i}^* - x_\mathbf{i}\|_F < \sqrt{2\eta\sigma_l}$ for all $l \in \{1, \ldots, L\}$, where $x_\mathbf{i}^* = \arg\min_{x_\mathbf{i}} E_{\theta^*}(x_\mathbf{i}, \mathbf{z}_\mathbf{i})$, so that $\nabla_{x_\mathbf{i}^*} E_{\theta^*}(x_\mathbf{i}^*, \mathbf{z}_\mathbf{i}) = 0$, and η represents the residual error after optimization.

Therefore, instead of directly predicting a single most probable tensor from the observations, our method enables the model to capture the dependency between tensor entry $x_\mathbf{i}$ and its corresponding latent factor $\mathbf{z}_\mathbf{i}$, which act as conditions. Through score matching, the model automatically assigns lower energy values to correct answers and higher energies to corrupted or noisy ones.

Network Architecture. Effective modeling of probabilistic manifolds relies heavily on the network architecture. We define the energy function as $E_\theta(\tilde{x}_\mathbf{i}, \mathbf{z}_\mathbf{i}) = g_{\theta_1}\big(g_2(g_{\theta_3}(\tilde{x}_\mathbf{i}), g_{\theta_4}(\mathbf{z}_\mathbf{i}))\big)$, where g_{θ_3} and g_{θ_4} are MLPs that encode $\tilde{x}_\mathbf{i}$ and $\mathbf{z}_\mathbf{i}$, respectively, and g_2 is a summation or concatenation layer that couples tensor values with latent factors. The output layer is parameterized by g_{θ_1}. To model dynamic tensors, we incorporate a sinusoidal positional encoding layer (Tancik et al. 2020), denoted $\gamma_{\theta_t}(t_i)$, to capture temporal information. The energy function then becomes $E_\theta(\tilde{x}_\mathbf{i}, \mathbf{z}_\mathbf{i}, t_i) = g_{\theta_1}\big(g_2(g_{\theta_3}(\tilde{x}_\mathbf{i}), g_{\theta_4}(\mathbf{z}_\mathbf{i}), \gamma_{\theta_t}(t_i))\big)$. This architecture is widely used for temporal modeling and has been shown effective in capturing high-frequency patterns when combined with MLPs. In the above cases, the latent factor $\mathbf{z}_\mathbf{i}$ is learnable parameter. While processing visual data, we employ a sinusoidal positional encoding layer to predict the latent factors, denoted $\mathbf{z}_\mathbf{i} = \gamma_{\theta_\mathbf{z}}(\mathbf{i})$, and obtain an implicit neural representation.

Posterior Sampling. With the learned parameters θ and latent factors $\mathcal{Z}$, the $\mathcal{X}$-subproblem can be formulated as

$$\arg\min_{x_\mathbf{i}} E_\theta(x_\mathbf{i}, \mathbf{z}_\mathbf{i}), \forall \mathbf{i} \in \Omega. \quad (8)$$

Unlike traditional TDs, our method does not yield direct predictions even after learning the latent factors, due to its dependence on an energy-based formulation. Instead, we rely on sampling techniques to approximate the conditional distribution $p(x_\mathbf{i}|\mathbf{z}_\mathbf{i})$. As detailed in Algorithm 1, for continuous-valued data, we employ Annealed Langevin Dynamic (ALD) algorithm that leverages the learned score function. For discrete-valued data, such as image tensors, we instead apply a grid search algorithm to estimate the most probable values.

Algorithm 1: ALD algorithm for Tensor Recovery.

Input: Parameters $\theta^*, \mathcal{Z}, \{\sigma_l\}_{l=1}^L, K, \mathbf{i} \in \Omega$.

Initialize $x_{\mathbf{i}}$.
for $l \leftarrow 1$ to L **do**
  Get step size $\alpha_l \leftarrow \epsilon \cdot \sigma_l^2/\sigma_L^2$.
  for $k \leftarrow 1$ to K **do**
    Draw $\epsilon_k \sim \mathcal{N}(0,1)$
    $x_{\mathbf{i}} \leftarrow x_{\mathbf{i}} - \alpha_l \nabla_{\tilde{x}_{\mathbf{i}}} E_{\theta^*}(x_{\mathbf{i}}, \mathbf{z}_{\mathbf{i}}) + \sqrt{2\alpha_l}\,\epsilon_k$
  end for
end for

Output: $x_{\mathbf{i}}$

Algorithm 2: BCD Algorithm for Tensor Denoising

Input: Observed tensor $\hat{\mathcal{X}}$ and λ_S, T, Ω.

Initialize $\mathcal{X}^{(0)} \leftarrow \hat{\mathcal{X}}$
Initialize $\mathcal{S}^{(0)} \leftarrow \mathbf{0}$
for $t = 1, 2, \ldots, T$ **do**
  for $\mathbf{i} \in \Omega$ **do**
    Posterior sampling with grid search or Algorithm 1 $x_{\mathbf{i}}^{(t)} \leftarrow \arg\min_{x_{\mathbf{i}}} E_\theta(x_{\mathbf{i}}, \mathbf{z}_{\mathbf{i}})$.
    $s_{\mathbf{i}}^{(t)} \leftarrow \text{Soft}_{\lambda_S/2}(\hat{x}_{\mathbf{i}} - x_{\mathbf{i}}^{(t)})$
    $x_{\mathbf{i}}^{(t)} \leftarrow \hat{x}_{\mathbf{i}} - s_{\mathbf{i}}^{(t)}$
    Update θ and $\mathbf{z}_{\mathbf{i}}$ by minimizing $\mathcal{L}_D$ in Eq. (7) and $\mathcal{L}_S$ in Eq. (11) with Adam optimizer.
  end for
end for

Output: $\mathcal{X}^* \leftarrow fold(\{x_{\mathbf{i}}^{(T)}\}), \mathcal{S}^* \leftarrow fold(\{s_{\mathbf{i}}^{(T)}\})$

Smooth and Sparse Regularization for Denoising from Energy Perspective

Given the estimation of energy function $E_\theta(\cdot,\cdot)$ and the learned latent factor $\mathcal{Z}$, we aim to obtain the complete tensor, conditional on the observed values, i.e., $p(\mathcal{X}, \mathcal{S}|\hat{\mathcal{X}}, \mathcal{Z})$. The estimated complete tensor $\mathcal{X}$ and noise $\mathcal{S}$ can be updated by taking block coordinate descent algorithm. Using Bayesian rule, this problem can be formulated as

$$p(\mathcal{X}, \mathcal{S}|\hat{\mathcal{X}}, \mathcal{Z}) \propto p(\hat{\mathcal{X}}|\mathcal{X}, \mathcal{S}) p(\mathcal{S}) p(\mathcal{X}|\mathcal{Z}). \tag{9}$$

The above equation holds because latent factor $\mathcal{Z}$ is independent of noise $\mathcal{S}$ and observation $\hat{\mathcal{X}}$. With the probability relationship, i.e., $p(\hat{\mathcal{X}}|\mathcal{X}, \mathcal{S}) \propto \exp\left(-\frac{\|\hat{\mathcal{X}} - \mathcal{X} - \mathcal{S}\|_F^2}{2\sigma^2}\right)$, and the sparse noise prior $p(\mathcal{S}) \propto \exp\left(-\lambda \|\mathcal{S}\|_{\ell_1}\right)$, and we solve the above problem by minimizing the negative log-posterior. Thus the $\mathcal{S}$-subproblem can be formulated as

$$\arg\min_{\mathcal{S}} \|\hat{\mathcal{X}} - \mathcal{X} - \mathcal{S}\|_F^2 + \lambda_S \|\mathcal{S}\|_{\ell_1}. \tag{10}$$

which can be exactly solved by element-wise soft-thresholding operator, i.e., $\mathcal{S} = Soft_{\lambda_S/2}(\hat{\mathcal{X}} - \mathcal{X})$, where the $Soft_{\frac{\lambda_S}{2}}(\cdot) = sgn(\cdot)\max(|\cdot| - \frac{\lambda_S}{2}, 0)$.

It is challenging to effectively remove complex mixed noise patterns such as Gaussian noise, stripe noise, and dead lines using only sparse regularization. Smoothness regularization has proven effective for denoising tasks, with total variation (TV) loss as a representative example. However, reconstructing the entire tensor in each iteration is computationally expensive and limits scalability. To address this issue while promoting spatial smoothness and improving generalization, we propose an energy-based smoothing loss that penalizes large variations in the energy function with respect to small perturbations in location:

$$\mathcal{L}_S(\theta; t) = \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \sigma_S^2 \mathbf{I}), \mathbf{i} \sim \Omega} \left[E_\theta \left(x_{\mathbf{i}}^{(t)}, \gamma_{\theta_{\mathbf{z}}}(\mathbf{i} + \epsilon) \right) \right], \tag{11}$$

where ϵ denotes a random perturbation sampled from an isotropic Gaussian distribution, $x_{\mathbf{i}}^{(t)}$ is the posterior sample obtained at iteration t, as defined in Eq. (8), and $\mathbf{z}_{\mathbf{i}+\epsilon} = \gamma_{\theta_{\mathbf{z}}}(\mathbf{i} + \epsilon)$ represents the perturbed latent factor. This formulation encourages the model to assign lower energy values to the joint configuration of the central entry and its neighboring latent factors, thereby promoting local smoothness. It significantly reduces computational overhead while maintaining strong denoising performance. For clarity, we summarize the steps of our proposed BCD algorithm for tensor denoising in Algorithm 2.

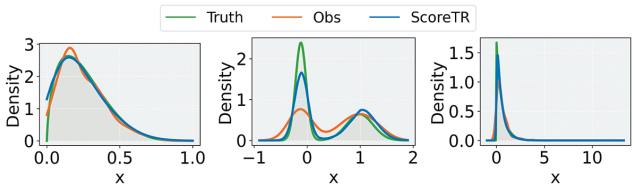


Figure 2: Simulation results for different distributions with Beta, MoG and Exponential are presented respectively.

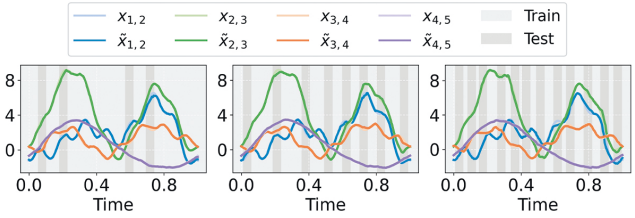


Figure 3: Simulation results for continuous tensor recovery with MR = 0.2, 0.4, and 0.6 are presented respectively.

4 Experiments

This section presents the experimental results of our method, in comparison with state-of-the-art approaches, evaluated on both synthetic and real-world tensors. The implementation details, ablation experiments and visualizations are all in the appendix.

Simulation study

Tensor with Non-Gaussian Distribution. Traditional TD methods often assume that tensor entries follow a Gaussian distribution. However, real-world data frequently exhibit more complex distributions. In this experiment, we evaluate the ability of our model to capture non-Gaussian distributions. We consider a two-mode tensor of size $I \times I$,

with $I = 8$, and generate two latent factor matrices of size $I \times R$, where the rank R is set to 5. Then, conditioned on these latent factors, we generate tensor observations from three different non-Gaussian distributions: (1) Beta distribution, (2) Mixture of Gaussians (MoG), and (3) Exponential distribution. For each entry, we generate $N = 200$ independent samples. For all settings, the latent factors are sampled independently and identically from a uniform distribution: $\mathbf{Z}^1, \mathbf{Z}^2 \overset{\text{i.i.d.}}{\sim} \text{Uni}(0, 1)$. For Beta distribution, each entry is sampled as $x_{ij} \overset{\text{i.i.d.}}{\sim} \text{Beta}\left((\mathbf{Z}^1\mathbf{Z}^{2,\top})_{ij},\, 5\right)$. For Mixture of Gaussians (MoG), each entry follows $x_{ij} \overset{\text{i.i.d.}}{\sim} 0.6 \cdot \mathcal{N}\left(\cos((\mathbf{Z}^1\mathbf{Z}^{2,\top})_{ij}),\, 0.1^2\right) + 0.4 \cdot \mathcal{N}\left(\sin((\mathbf{Z}^1\mathbf{Z}^{2,\top})_{ij}),\, 0.25^2\right)$. For Exponential distribution, each entry is sampled as $x_{ij} \overset{\text{i.i.d.}}{\sim} \text{Exp}\left((\mathbf{Z}^1\mathbf{Z}^{2,\top})_{ij}\right)$.

We visualize the learned probability density function of a single tensor entry, as shown in Figure 2. Our method does not simply fit the observed values, instead, it provides a more accurate approximation of the underlying true distribution. This demonstrates the flexibility of our model in capturing complex, non-Gaussian data distributions.

Continuous Tensor. For a continuous-time tensor, where each entry corresponds to a time series. Following the setup in (Tao, Tanaka, and Zhao 2023), we use a two-mode tensor of size 8×8, with each entry being a time series of length 200. We first generate latent factor matrices of size 8×2. Each row of the first latent factor matrix is sampled independently as $\mathbf{z}_\mathbf{i}^1 \sim \mathcal{N}([0, 2], 2 \cdot \mathbf{I})$, and each row of the second latent factor matrix is sampled as $\mathbf{z}_\mathbf{i}^2 \sim \mathcal{N}([1, 1], 2 \cdot \mathbf{I})$. Then, for each entry, we generate $N = 200$ observations over the time interval $t \in [0, 1]$. The tensor entries are computed using a weighted sum of temporal basis functions: $x_\mathbf{i}(t) = \sum_{r_1=1}^{2} \sum_{r_2=1}^{2} z_{i_1 r_1}^1 z_{i_2 r_2}^2 \omega_{r_1 r_2}(t)$, where $\omega_{11}(t) = \sin(2\pi t)$, $\omega_{12}(t) = \cos(2\pi t)$, $\omega_{21}(t) = \sin^2(2\pi t)$, and $\omega_{22}(t) = \cos(5\pi t)\sin^2(5\pi t)$. This synthetic dataset incorporates both low- and high-frequency fluctuations to mimic realistic temporal dynamics. We evaluate performance at missing rates (MR) of 20%, 40%, and 60% by randomly selecting 4, 8, and 12 start points per entry and masking the subsequent 5% of timestamps as missing.

Figure 3 shows the completion results, plotting the learned trajectories of four tensor entries. It can be observed that higher missing rates lead to a slight performance decline. Our method is capable of adapting to complex scenarios that involve both low-frequency trends and high-frequency components, achieving near-perfect predictions.

Tensor Completion

We evaluate our model on tensor completion tasks using two sparse and two dynamic tensors.

Sparse Tensor Completion. We evaluate our model on two sparsely observed tensor datasets (Zhe et al. 2015): (1) *Alog*, a file access log dataset with dimensions 200 users $\times$ 100 actions $\times$ 200 resources, containing approximately 0.33% nonzero entries; and (2) *ACC*, a three-way tensor derived from a code repository management system with dimensions 3k users $\times$ 150 actions $\times$ 30k resources, containing approximately 0.009% nonzero entries. We follow the same data split as in (Tao, Tanaka, and Zhao 2023) and report results based on 5-fold cross-validation.

Metric	RMSE				MAE			
Method	R3	R5	R8	R10	R3	R5	R8	R10
Alog ($200 \times 100 \times 200$)								
CP-WOPT	1.486	1.386	1.228	1.355	0.694	0.664	0.610	0.658
GPTF	0.911	0.867	0.878	0.884	0.511	0.494	0.530	0.554
HGP-GPTF	0.896	0.867	0.850	0.844	0.479	0.473	0.474	0.480
POND	0.885	0.871	0.858	0.857	0.463	0.454	0.444	0.443
CosTCo	0.999	0.936	0.930	0.909	0.523	0.481	0.514	0.481
EnergyTD	0.864	0.835	0.840	0.833	0.450	0.433	0.424	0.409
ScoreTR	**0.845**	**0.824**	**0.813**	**0.811**	**0.409**	**0.397**	**0.391**	**0.387**
Acc ($3k \times 150 \times 30k$)								
CP-WOPT	0.533	0.592	0.603	0.589	0.138	0.147	0.148	0.147
GPTF	0.367	0.357	0.359	0.368	0.152	0.150	0.167	0.182
HGP-GPTF	0.355	0.344	0.341	0.338	0.125	0.129	0.139	0.145
CosTCo	0.385	0.376	0.363	0.348	0.117	0.137	0.107	0.101
EnergyTD	0.348	0.336	0.328	0.328	0.110	0.101	0.094	0.101
ScoreTR	**0.325**	**0.319**	**0.322**	**0.320**	**0.076**	**0.073**	**0.073**	**0.074**

Table 1: Results of sparse tensor completion.

We compare our method against six baseline models: (1) CP-WOPT (Bader and Kolda 2008), a CP decomposition method with stochastic optimization; (2) GPTF (Zhe et al. 2016), a Gaussian process-based tensor factorization using stochastic variational inference; (3) HGP-GPTF (Tillinghast, Wang, and Zhe 2022), an extension of GPTF with a hierarchical Gamma process prior; (4) POND (Tillinghast et al. 2020), a probabilistic non-linear TD using deep kernels with convolutional neural networks; (5) CoSTCo (Liu et al. 2019), a non-linear TD that employs CNNs to map latent factors to tensor entries; and (6) EnergyTD (Tao, Tanaka, and Zhao 2023), an undirected graphical TD model solved via variational conditional noise-contrastive estimation, which is the SOTA method for sparse tensor completion. For all methods, we evaluate ranks $R \in \{3, 5, 8, 10\}$.

The results are presented in Table 1, where the root mean square error (RMSE) and mean absolute error (MAE) are reported as averages. Results for POND are omitted on the ACC dataset due to its high computational cost and excessive memory requirements. Our model achieves significant performance gains over the current state-of-the-art method across all cases. The energy-based methods demonstrate notable superiority over those relying on prior contraction rules, highlighting the benefits of adopting more flexible modeling frameworks beyond the Dirac delta distribution.

Continuous Tensor Completion. We evaluate our model on two continuous-time tensor datasets: (1) *Air*, the Beijing air quality dataset (Zhang et al. 2017), with dimensions 12 sites $\times$ 6 pollutants and approximately 1×10^4 observations across different time stamps; and (2) *Click*, an ad click-through dataset (Wang and Zhe 2022), with dimensions 7 banner positions $\times$ 2842 site domains $\times$ 4127 mobile apps, containing approximately 5×10^4 entries at varying time

Metric	RMSE				MAE			
Method	R3	R5	R8	R10	R3	R5	R8	R10
Air ($12 \times 6 \times T$)								
CTCP	1.020	1.022	1.022	1.022	0.784	0.785	0.787	0.787
CTGP	0.475	0.463	0.459	0.458	0.318	0.304	0.301	0.299
CTNN	1.013	1.005	0.999	1.013	0.780	0.777	0.776	0.780
NNDTN	0.377	0.364	0.334	0.328	0.247	0.239	0.217	0.212
NONFAT	0.339	0.335	0.351	0.342	0.224	0.219	0.228	0.223
THIS-ODE	0.569	0.566	0.542	0.541	0.415	0.409	0.395	0.391
EnergyTD	0.302	0.291	0.300	0.283	0.184	0.177	0.172	0.184
ScoreTR	**0.242**	**0.259**	**0.259**	**0.240**	**0.156**	**0.170**	**0.168**	**0.154**
Click ($7 \times 2842 \times 4127 \times T$)								
CTCP	2.063	2.020	2.068	2.009	1.000	0.977	1.005	0.969
CTGP	1.424	1.423	1.404	1.392	0.880	0.877	0.856	0.849
CTNN	1.820	1.820	1.820	1.820	1.077	1.053	1.083	1.071
NNDTN	1.418	1.409	1.407	1.410	0.858	0.856	0.859	0.863
NONFAT	1.400	1.411	1.365	1.351	0.853	0.873	0.832	0.812
THIS-ODE	1.421	1.413	1.408	1.395	0.836	0.836	0.832	0.829
EnergyTD	1.396	1.385	1.356	1.357	0.777	0.775	0.772	0.773
ScoreTR	**1.374**	**1.368**	**1.355**	**1.346**	**0.746**	**0.749**	**0.744**	**0.747**

Table 2: Results of continuous-time tensor completion.

stamps. We follow the same data split as in (Wang and Zhe 2022) and report results based on 5-fold cross-validation.

We compare with the following continuous-time tensor modeling approaches: (1) Nonparametric factor trajectory learning (NONFAT) (Wang and Zhe 2022); (2) Continuous-time CP (CTCP) (Zhang et al. 2021), which models the temporal dynamics of CP coefficients using polynomial splines; (3) Continuous-time GP (CTGP), an extension of GPTF (Zhe et al. 2016) that incorporates time stamps into Gaussian process kernels; (4) Continuous-time NN decomposition (CTNN), a variant of CoSTCo (Liu et al. 2019) that uses time stamps as inputs to learn continuous latent trajectories; (5) Discrete-time NN decomposition with non-linear dynamics (NNDTN) (Wang and Zhe 2022), which employs RNN-based dynamics to model temporal evolution; (6) Tensor high-order interaction learning via ODEs (THIS-ODE) (Li, Kirby, and Zhe 2022), which captures continuous-time tensor entry trajectories using neural ODEs; and (7) EnergyTD (Tao, Tanaka, and Zhao 2023), an undirected graphical tensor decomposition model solved via variational conditional noise-contrastive estimation, which is the state-of-the-art method on continuous-time tensor completion.

The results are presented in Table 2, with the RMSE and MAE reported as averages. Our model significantly outperforms the current state-of-the-art method across cases. Most methods rely on minimizing squared loss under a Gaussian process assumption on the temporal data. In contrast, our model makes no such assumption, enabling more flexible adaptation to the underlying data distribution. By modifying the network architecture to incorporate side information, our approach improves scalability. As an energy-based method, our model learns gradients under different signal-to-noise ratios, leading to better performance than EnergyTD, which performs anti-noise training in a single stage.

MSIs *Balloons, Beads, Flowers, Fruits* ($512 \times 512 \times 31$)						
Method	PSNR	SSIM	NRMSE	PSNR	SSIM	NRMSE
Noise	Case 1			Case 2		
Observed	16.22	0.084	0.902	16.26	0.109	0.900
M^2DMT	29.80	0.720	0.158	30.94	0.748	0.136
LRTC-ENR	31.26	0.756	0.152	33.88	0.845	0.127
HLRTF	30.57	0.731	0.159	32.75	0.781	0.152
DeepTensor	29.97	0.725	0.155	31.03	0.747	0.140
LRTFR	31.32	0.736	0.167	32.89	0.784	0.141
ScoreTR	**32.32**	**0.856**	**0.150**	**34.54**	**0.865**	**0.118**
Noise	Case 3			Case 4		
Observed	16.12	0.101	0.912	16.20	0.107	0.906
M^2DMT	30.91	0.747	0.170	28.32	0.731	0.179
LRTC-ENR	31.49	0.770	0.156	28.96	0.753	0.174
HLRTF	31.51	0.791	0.158	29.53	0.756	0.173
DeepTensor	30.79	0.786	0.169	29.77	0.741	0.179
LRTFR	31.96	0.794	0.153	31.27	0.776	0.162
ScoreTR	**32.51**	**0.870**	**0.144**	**31.83**	**0.835**	**0.154**
Noise	Case 5			Case 6		
Observed	16.07	0.101	0.917	17.79	0.290	0.760
M^2DMT	27.17	0.769	0.230	35.72	0.944	0.103
LRTC-ENR	28.07	0.780	0.216	39.66	0.968	0.076
HLRTF	27.93	0.774	0.228	37.81	0.967	0.081
DeepTensor	27.89	0.772	0.221	37.19	0.963	0.087
LRTFR	**29.97**	0.782	**0.187**	39.32	0.971	0.068
ScoreTR	29.71	**0.841**	0.195	**43.83**	**0.988**	**0.042**

Table 3: Results of multispectral image denoising.

Image Recovery

We evaluate our model on image inpainting and denoising tasks, and compare it with state-of-the-art low-rank tensor-based methods, including: (1) M^2DMT (Fan 2021), a fully nonlinear framework for multi-mode deep matrix and tensor factorization; (2) LRTC-ENR (Fan et al. 2023), which applies euclidean-norm-induced regularization based on the Schatten-p quasi-norm and is solved via L-BFGS (Liu and Nocedal 1989); (3) HLRTF (Luo et al. 2022), which incorporates a DNN into the t-SVD framework using parametric total variation regularization; (4) DeepTensor (Saragadam et al. 2024), which represents a tensor as the product of low-rank factors generated by deep neural networks; and (5) LRTFR (Luo et al. 2023), which models continuous representations as low-rank tensor functions using Tucker decomposition, and is currently the state-of-the-art method for image recovery.

Image Denoising. MSI denoising aims to recover a clean image from a noisy observation. In practice, MSIs are often degraded by mixed noise types, including Gaussian noise, sparse noise, etc. The experiments are conducted on four MSIs from the CAVE dataset[1]. According to the experimental setup of LRTFR(Luo et al. 2023), we consider six noise scenarios to evaluate the robustness and effectiveness of denoising algorithms under various noise conditions. (1) Case

[1]https://www.cs.columbia.edu/CAVE/databases/multispectral/

Sampling rate		0.1			0.15			0.2		
Data	Method	PSNR	SSIM	NRMSE	PSNR	SSIM	NRMSE	PSNR	SSIM	NRMSE
Color images *Sailboat* *House* *Peppers* *Plane* ($512 \times 512 \times 3$)	*Observed*	4.846	0.023	0.949	5.095	0.030	0.922	5.358	0.038	0.895
	M^2DMT	22.06	0.573	0.145	23.49	0.650	0.137	23.89	0.692	0.116
	LRTC-ENR	23.56	0.628	0.128	24.61	0.694	0.114	25.16	0.707	0.102
	HLRTF	22.49	0.540	0.136	24.41	0.679	0.110	25.39	0.711	0.097
	DeepTensor	21.50	0.484	0.150	24.53	0.682	0.118	26.31	0.717	0.101
	LRTFR	23.03	0.597	0.132	26.22	0.695	0.084	27.49	0.741	0.073
	ScoreTR	**25.34**	**0.755**	**0.092**	**27.22**	**0.804**	**0.075**	**28.14**	**0.829**	**0.068**
MSIs *Toys* *Flowers* ($512 \times 512 \times 31$)	*Observed*	13.96	0.386	0.949	14.21	0.418	0.922	14.47	0.447	0.894
	M^2DMT	34.89	0.910	0.107	36.82	0.928	0.092	38.19	0.934	0.082
	LRTC-ENR	35.91	0.928	0.094	37.14	0.935	0.080	39.33	0.950	0.070
	HLRTF	36.32	0.935	0.091	38.64	0.942	0.076	40.19	0.955	0.067
	DeepTensor	38.40	0.947	0.088	39.99	0.951	0.077	41.20	0.965	0.066
	LRTFR	40.16	0.969	0.047	42.74	0.982	0.035	44.28	0.985	0.029
	ScoreTR	**41.27**	**0.985**	**0.042**	**44.22**	**0.990**	**0.030**	**46.28**	**0.993**	**0.024**
Videos *Foreman* *Carphone* ($144 \times 176 \times 100$)	*Observed*	5.548	0.017	0.949	5.797	0.024	0.922	6.059	0.031	0.894
	M^2DMT	23.51	0.701	0.124	25.21	0.769	0.102	26.47	0.815	0.095
	LRTC-ENR	24.23	0.730	0.117	25.91	0.793	0.094	27.56	0.826	0.088
	HLRTF	24.66	0.768	0.104	26.49	0.830	0.085	28.10	0.837	0.071
	DeepTensor	25.67	0.813	0.114	27.34	0.855	0.080	28.89	0.851	0.067
	LRTFR	28.53	0.828	0.067	29.36	0.854	0.061	29.77	0.866	0.058
	ScoreTR	**33.35**	**0.945**	**0.039**	**35.04**	**0.958**	**0.032**	**36.30**	**0.966**	**0.028**

Table 4: Results of multidimensional images inpainting.

1 introduces Gaussian noise with a standard deviation of 0.2; (2) Case 2 combines Gaussian noise ($\sigma = 0.1$) with sparse noise at a sparsity rate of 0.1; (3) Case 3 extends Case 2 by adding dead lines across all spectral bands; (4) Case 4 includes the same noise as in Case 2 along with 10% stripe noise in 40% of the spectral bands; (5) Case 5 further incorporates dead lines into Case 4, combining stripe noise and dead lines; and (6) Case 6 considers only sparse noise with rate of 0.1. These scenarios provide a comprehensive benchmark for evaluating MSI denoising methods under diverse and realistic noise conditions.

The completion results are presented in Table 3, where the average values of PSNR, SSIM, and NRMSE are reported. Our model significantly outperforms the current state-of-the-art method in most cases, particularly in terms of SSIM. We incorporate the proposed smooth loss from an energy perspective into the block coordinate descent framework, enabling effective handling of the MSI denoising task. It has been shown that assigning lower energy to tensor values and their neighboring latent factors can achieve effective smooth regularization. This approach avoids the computational burden of TV regularization, which requires full tensor posterior sampling in each iteration and is therefore time-consuming.

Image Inpainting. We evaluate our model on three type of image data: (1) *RGB image*[2]; (2) *Multispectral image*[1]; and (3) *Video*[3]. We evaluated performance under random missing conditions with sampling rates (SRs) of 10%, 15%, 20%.

[2]https://sipi.usc.edu/database/database.php

[3]http://trace.eas.asu.edu/yuv/

The completion results are presented in Table 4, where the peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), and normalized root mean square error (NRMSE) are reported as averages. Our model significantly outperforms the current state-of-the-art method across all test cases, with particularly notable improvements on video data, demonstrating its strong performance on visual information. The superior performance of our method can be attributed to its ability to jointly encode low-rankness and smoothness into the learned representation. Traditional methods heavily rely on predefined contraction structures and rank definitions, making them less adaptable to tensors from diverse domains, as the intrinsic structures of such tensors often vary significantly. In contrast, our approach is capable of handling tensor data from multiple fields simultaneously, without being constrained by prior contraction rules.

5 Conclusion

We introduce an innovative low-rank tensor recovery model solved via denoising score matching, distinguished by its flexibility in adapting to diverse structures and distributions. Our method eliminates the need for computing expensive high-dimensional tensor contractions, thereby overcoming the limitations of traditional approaches that rely on predefined contraction rules. By learning the gradient of the joint probability distribution of tensor entries and latent factors, it achieves superior performance over the state-of-the-art methods across multiple domains and tasks.

Acknowledgments

This research was supported by the National Science and Technology Major Project (2025ZD1401102), and the Innovation Foundation for Doctor Dissertation of Northwestern Polytechnical University (CX2025074).

References

Bader, B. W.; and Kolda, T. G. 2008. Efficient MATLAB computations with sparse and factored tensors. *SIAM Journal on Scientific Computing*, 30(1): 205–231.

Chen, R.; Yang, D.; and Zhang, C.-H. 2022. Factor models for high-dimensional tensor time series. *Journal of the American Statistical Association*, 117(537): 94–116.

Chu, W.; and Ghahramani, Z. 2009. Probabilistic models for incomplete multi-dimensional arrays. In *Artificial Intelligence and Statistics*, 89–96. PMLR.

Dai, W.; Fan, J.; Miao, Y.; and Hwang, K. 2023. Deep learning model compression with rank reduction in tensor decomposition. *IEEE Transactions on Neural Networks and Learning Systems*.

Fan, J. 2021. Multi-mode deep matrix and tensor factorization. In *international conference on learning representations*.

Fan, J.; Ding, L.; Yang, C.; Zhang, Z.; and Udell, M. 2023. Euclidean-Norm-Induced Schatten-p Quasi-Norm Regularization for Low-Rank Tensor Completion and Tensor Robust Principal Component Analysis. *Transactions on Machine Learning Research*.

Fang, S.; Wang, Z.; Pan, Z.; Liu, J.; and Zhe, S. 2021. Streaming Bayesian deep tensor factorization. In *International conference on machine learning*, 3133–3142. PMLR.

Giampouras, P.; Vidal, R.; Rontogiannis, A.; and Haeffele, B. 2020. A novel variational form of the Schatten-p quasi-norm. *Advances in Neural Information Processing Systems*, 33: 21453–21463.

Glasser, I.; Sweke, R.; Pancotti, N.; Eisert, J.; and Cirac, I. 2019. Expressive power of tensor-network factorizations for probabilistic modeling. *Advances in neural information processing systems*, 32.

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

Kuznetsov, M.; Polykovskiy, D.; Vetrov, D. P.; and Zhebrak, A. 2019. A prior of a googol gaussians: a tensor ring induced prior for generative models. *Advances in Neural Information Processing Systems*, 32.

Li, C.; and Sun, Z. 2020. Evolutionary topology search for tensor network decomposition. In *International conference on machine learning*, 5947–5957. PMLR.

Li, C.; Zeng, J.; Tao, Z.; and Zhao, Q. 2022. Permutation search of tensor network structures via local sampling. In *International conference on machine learning*, 13106–13124. PMLR.

Li, S.; Kirby, R.; and Zhe, S. 2022. Decomposing temporal high-order interactions via latent odes. In *International Conference on Machine Learning*, 12797–12812. PMLR.

Liu, D. C.; and Nocedal, J. 1989. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1): 503–528.

Liu, H.; Li, Y.; Tsang, M.; and Liu, Y. 2019. Costco: A neural tensor completion model for sparse tensors. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 324–334.

Luo, Y.; Zhao, X.; Li, Z.; Ng, M. K.; and Meng, D. 2023. Low-rank tensor function representation for multi-dimensional data recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Luo, Y.; Zhao, X.-L.; Meng, D.; and Jiang, T.-X. 2022. HLRTF: Hierarchical low-rank tensor factorization for inverse problems in multi-dimensional imaging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19303–19312.

Miller, J.; Rabusseau, G.; and Terilla, J. 2021. Tensor networks for probabilistic sequence modeling. In *International Conference on Artificial Intelligence and Statistics*, 3079–3087. PMLR.

Novikov, A.; Podoprikhin, D.; Osokin, A.; and Vetrov, D. P. 2015. Tensorizing neural networks. *Advances in neural information processing systems*, 28.

Oseledets, I. V. 2011. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5): 2295–2317.

Rai, P.; Wang, Y.; Guo, S.; Chen, G.; Dunson, D.; and Carin, L. 2014. Scalable Bayesian low-rank decomposition of incomplete multiway tensors. In *International Conference on Machine Learning*, 1800–1808. PMLR.

Saragadam, V.; Balestriero, R.; Veeraraghavan, A.; and Baraniuk, R. G. 2024. DeepTensor: Low-rank tensor decomposition with deep network priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Song, Y.; and Ermon, S. 2019. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32.

Tancik, M.; Srinivasan, P.; Mildenhall, B.; Fridovich-Keil, S.; Raghavan, N.; Singhal, U.; Ramamoorthi, R.; Barron, J.; and Ng, R. 2020. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33: 7537–7547.

Tao, Z.; Tanaka, T.; and Zhao, Q. 2023. Undirected probabilistic model for tensor decomposition. *Advances in Neural Information Processing Systems*, 36: 25837–25853.

Tillinghast, C.; Fang, S.; Zhang, K.; and Zhe, S. 2020. Probabilistic neural-kernel tensor decomposition. In *2020 IEEE International Conference on Data Mining (ICDM)*, 531–540. IEEE.

Tillinghast, C.; Wang, Z.; and Zhe, S. 2022. Nonparametric sparse tensor factorization with hierarchical Gamma processes. In *International Conference on Machine Learning*, 21432–21448. PMLR.

Vincent, P. 2011. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7): 1661–1674.

Wang, Z.; and Zhe, S. 2022. Nonparametric factor trajectory learning for dynamic tensor decomposition. In *International Conference on Machine Learning*, 23459–23469. PMLR.

Wu, Z.-C.; Huang, T.-Z.; Deng, L.-J.; Dou, H.-X.; and Meng, D. 2022. Tensor wheel decomposition and its tensor completion application. *Advances in Neural Information Processing Systems*, 35: 27008–27020.

Xu, Z.; Yan, F.; and Qi, Y. 2012. Infinite tucker decomposition: nonparametric Bayesian models for multiway data analysis. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, 1675–1682.

Zhang, S.; Guo, B.; Dong, A.; He, J.; Xu, Z.; and Chen, S. X. 2017. Cautionary tales on air-quality improvement in Beijing. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2205): 20170457.

Zhang, Y.; Bi, X.; Tang, N.; and Qu, A. 2021. Dynamic tensor recommender systems. *Journal of machine learning research*, 22(65): 1–35.

Zhao, Q.; Zhang, L.; and Cichocki, A. 2015. Bayesian CP factorization of incomplete tensors with automatic rank determination. *IEEE transactions on pattern analysis and machine intelligence*, 37(9): 1751–1763.

Zhao, Q.; Zhou, G.; Xie, S.; Zhang, L.; and Cichocki, A. 2016. Tensor ring decomposition. *arXiv preprint arXiv:1606.05535*.

Zhe, S.; Xu, Z.; Chu, X.; Qi, Y.; and Park, Y. 2015. Scalable nonparametric multiway data analysis. In *Artificial intelligence and statistics*, 1125–1134. PMLR.

Zhe, S.; Zhang, K.; Wang, P.; Lee, K.-c.; Xu, Z.; Qi, Y.; and Ghahramani, Z. 2016. Distributed flexible nonlinear tensor factorization. *Advances in neural information processing systems*, 29.

Zheng, Y.-B.; Huang, T.-Z.; Zhao, X.-L.; Zhao, Q.; and Jiang, T.-X. 2021. Fully-Connected Tensor Network Decomposition and Its Application to Higher-Order Tensor Completion. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12): 11071–11078.

Zhou, G.; Cichocki, A.; Zhao, Q.; and Xie, S. 2015. Efficient nonnegative tucker decompositions: Algorithms and uniqueness. *IEEE Transactions on Image Processing*, 24(12): 4990–5003.