EAI Endorsed Transactions
on AI and Robotics

Research Article EAI.EU

A Huber-Regularized Latent Factorization of Tensors Approach for Large-Scale Incomplete Water Quality Data Imputation

Hengrui Tu[1], Xuke Wu[1,2,*], Tinghui Chen[2,3], Zhibin Li[3], Jinli Li[1]

[1]School of Computer and Software Engineering, Xihua University, Chengdu 610039, China

[2]Dazhou Key Laboratory of Government Data Security, Sichuan University of Arts and Science, Dazhou 635000, China

[3]School of Software Engineering, Chengdu University of Information Technology, Chengdu 610225, China

Abstract

An online water quality monitoring network continuously collects multiple key water quality indicators across distributed monitoring sites via wireless sensors, providing essential data support for intelligent water resource management and ecological assessment. These large-scale observations can be naturally represented as a high-dimensional and incomplete (HDI) water quality tensor with frequent missing entries. Such an HDI tensor contains rich multivariate, spatial, and temporal dependencies that can be effectively exploited for missing data recovery. Latent factorization of tensors (LFT) models have demonstrated remarkable capability in extracting latent knowledge from HDI tensors, thereby enabling accurate missing data imputation. However, existing LFT models generally adopt the conventional L_2 regularization scheme, making them highly sensitive to sensor outliers that are ubiquitous in real-time monitoring environments. To address this issue, this study proposes a Huber-regularized latent factorization of tensors (HRL) model under the Tucker decomposition framework. Specifically, a Huber regularization scheme is incorporated into the learning objective to improve robustness against outliers while preserving the smoothness of latent factor learning. Empirical studies on eight practical HDI water quality tensors demonstrate that the proposed HRL model consistently outperforms state-of-the-art models in terms of computational efficiency and imputation accuracy, offering an effective and robust solution for large-scale incomplete water quality data reconstruction and facilitating reliable downstream decision-making.

Received on 29 July 2026; accepted on 11 August 2026; published on 18 Augut 2026

Keywords: Water Quality Data Imputation, High-Dimensional and Incomplete Tensor, Tucker Decomposition, Latent Factorization of Tensors, Huber Regularization

doi:10.4108/airo.14233

1. Introduction

Water quality data serve as a critical basis for water resource management, pollution source tracing, and ecological assessment. In recent years, real-time monitoring sensor networks have been widely deployed in lakes, rivers, and reservoirs, continuously collecting multiple water quality indicators such as water temperature, pH, dissolved oxygen, conductivity, turbidity, permanganate index, ammonia nitrogen, total phosphorus, total nitrogen, chlorophyll-a, and algal density [1, 2]. However, due to sensor failures, equipment maintenance, and communication interruptions, substantial missing values commonly exist in monitoring systems, causing the collected data to exhibit a high-dimensional and incomplete (HDI) nature [3, 4]. Large-scale data not only reduce data availability but also severely compromise subsequent statistical analyses and modeling. Therefore, how to accurately impute these missing values has become an urgent research problem [5, 6].

Existing missing value imputation methods can be broadly categorized into interpolation-based, statistical,

*Corresponding author. Email: xuke_wu@163.com

and machine learning-based approaches[7]. Among them, latent factor analysis (LFA) models have attracted considerable attention due to their efficiency in handling HDI matrices. However, traditional LFA models typically treat different water quality variables, such as dissolved oxygen and ammonia nitrogen, as independent matrices and process them separately, thereby neglecting the intrinsic correlations among variables [8, 9]. In contrast, an HDI tensor can completely retain the structural, temporal and spatial patterns hidden in water quality data [10, 11]. Thus, latent factorization of tensors (LFT) models for water quality representation learning have gradually gained attention recently [1, 2, 42]. A key advantage of the LFT models is that they construct the learning objective solely based on the known entries of the original tensor, which naturally grants them scalability and computational efficiency when analyzing HDI tensors [12, 13]. Representative models in this category include the fused Candecomp/Parafac (CP) decomposition model [14], the nonnegative CP decomposition model [15], the bias-integrated adaptive LFT model [16], and the CP weighted optimization model [17].

When performing an LFT model, Tucker decomposition (TD) explicitly models interactions between latent factors (LFs) through a core tensor, providing potentially excellent representation ability [21]. Consequently, researchers have developed various LFT based on TD (LFT-TD) models [18, 19].These studies prove that incorporating custom-designed regularization schemes into LFT-TD models exhibits flexibility and accuracy in modeling intricate tensor structural and spatiotemporal patterns. Nevertheless, existing LFT-TD models typically employ L_2 penalty regularization scheme to enhance their generalization [20, 21]. The quadratic nature of the L_2 penalty makes it highly sensitive to outliers, so that even a small number of anomalous observations can severely bias the estimates. Directly replacing L_2 with L_1 regularization can enhance robustness, but its non-smoothness at the origin inevitably introduces numerical oscillations during gradient updates. An HDI water quality tensor is frequently contaminated by instrumental noise, sensor drift, and transient environmental disturbances, inevitably containing a substantial fraction of outliers. This makes it difficult for a single-scheme regularization to simultaneously achieve smoothness and outlier resistance. The Huber function behaves quadratically for small parameter values, preserving smoothness, and linearly for large values, suppressing the influence of outliers. It thus combines the advantages of both L_2 and L_1 regularizations [22, 23].

Inspired by the above discovery, we propose a Huber-regularized latent factorization of tensors model (HRL) based on the LFT-TD framework. It adopts Tucker decomposition as its backbone and explicitly introduces linear biases along the variable, site, and time dimensions to capture fluctuations hidden in HDI water quality tensors. Meanwhile, it applies Huber regularization uniformly to the core tensor, factor matrices, and bias terms, striking a balance between smoothness and outlier resistance. Specifically, we develop a normalized Huber scheme in which the linear regime has unit slope, with the transition between quadratic and linear behavior governed by a threshold parameter. Moreover, the model parameters are optimized via stochastic gradient descent (SGD), which computes the gradient and updates the parameters using only a single training sample at each step, resulting in low memory overhead and natural suitability for large-scale sparse tensor data. The main contributions of this work are as follows:

- An HRL model. It performs LFs on HDI water quality tensors with high imputation accuracy and computational efficiency.
- Detailed algorithm design and complexity analysis of the HRL model. It provides practical guidance for researchers to implement HRL for HDI tensor representation.

Extensive empirical studies on eight large-scale real-world water quality datasets covering multiple basins and varying data densities demonstrate that the HRL model outperforms state-of-the-art imputation models in both the accuracy and efficiency of missing value estimation for HDI tensors.

The remainder of this paper is organized as follows: Section 2 introduces preliminary knowledge. Section 3 elaborates on the HRL model. Section 4 reports the experimental results. Section 5 concludes the paper.

2. Preliminaries

2.1. Notations

Table 1 summarizes the primary symbols used throughout this paper. A variable-site-time third-order tensor is adopted as the foundational data structure for water quality modeling and is defined as:

Definition 1 (HDI Variable–Site–Time Tensor). Let I, J, and K denote the index sets of variables, sites, and time points, respectively. Define a third-order tensor $\mathbf{Y} \in \mathbb{R}^{|I|\times|J|\times|K|}$, whose entry y_{ijk} represents the observed value of variable $i \in I$ at site $j \in J$ and time $k \in K$. Let Λ and Γ denote the sets of observed and unobserved entries, respectively. If the proportion of missing entries is extremely high, i.e., $|\Gamma| \gg |\Lambda|$,then $\mathbf{Y}$ is called an HDI tensor.

Owing to the inevitable missing values and complex spatiotemporal correlations in water quality monitoring, such data can be organized into a partially observed third-order tensor, as shown in Figure 1. Observe that, in

Table 1. Key notations and descriptions

Symbol	Description
$\mathbf{Y}$	Target third-order variable-site-time HDI tensor.
$\mathbf{X}$	Rank-one tensor constructed from LF matrices.
$\hat{\mathbf{Y}}$	Low-rank approximation of $\mathbf{Y}$.
$\hat{y}_{ijk}$, y_{ijk}	Individual entries in $\hat{\mathbf{Y}}$ and $\mathbf{Y}$, respectively.
I, J, K	Variable, site, and time slot sets.
P, Q, R	The feature dimension of LF matrices.
$\mathbf{G}$	Core tensor in Tucker decomposition, $\mathbf{G} \in \mathbb{R}^{P\times Q\times R}$.
g_{pqr}	(p,q,r)-th entry of $\mathbf{G}$.
V, S, T	Variable, site, and time slot LF matrices.
$\boldsymbol{v}_p$, $\boldsymbol{s}_q$, $\boldsymbol{t}_r$	The p-th, q-th, and r-th LF vectors of variable, site, and time.
v_{ip}, s_{jq}, t_{kr}	Individual LFs.
$\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$	Bias vectors for variable, site, and time modes.
λ_g, λ_f, λ_b	Regularization coefficients for $\mathbf{G}$, {V, S, T}, and $\{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}\}$.
κ	A positive threshold.
τ	A positive smooth constant.
η	Learning rate for SGD.
Γ, Λ	Unknown and known entry sets of $\mathbf{Y}$.
Φ, Ω, Ψ	Training, validation, and test sets.
$\mathbb{R}$	Real number domain.
$\circ$	Outer product of two vectors.
$\lvert\cdot\rvert$	Cardinality of a set.

this paper, the time dimension is referred to as a discrete set of time slices unless otherwise indicated.

2.2. LFT-TD

This work employs TD to LFT on the target tensor [24]. While CP decomposition pursues a parsimonious representation, the difficulty of obtaining the optimal solution is excessively high. In contrast, TD gains numerical stability and structural flexibility through an auxiliary core tensor that arbitrates the cross-mode interactions among latent features [25]. Concretely, the target tensor $\mathbf{Y}$ is reconstructed as

$$\mathbf{Y} \approx \mathbf{G} \times_v \mathrm{V} \times_s \mathrm{S} \times_t \mathrm{T}, \tag{1}$$

where $\mathrm{V} \in \mathbb{R}^{|I|\times P}$, $\mathrm{S} \in \mathbb{R}^{|J|\times Q}$, and $\mathrm{T} \in \mathbb{R}^{|K|\times R}$ denote the latent factor matrices governing the variable, site, and time modes, respectively. The core tensor $\mathbf{G} \in \mathbb{R}^{P\times Q\times R}$ captures the interaction levels among different latent features. The symbols $\times_v$, $\times_s$, and $\times_t$ signify the corresponding modal products along the three feature dimensions [26]. To dissect these intricate interactions in greater detail, a rank-one tensor formulation is introduced below.

Definition 2 (Rank-One Tensor). A tensor $\mathbf{X}_{pqr} \in \mathbb{R}^{|I|\times|J|\times|K|}$ is a rank-one tensor if it can be expressed as the outer product of three latent factor vectors $\boldsymbol{v}_p$, $\boldsymbol{s}_q$, and $\boldsymbol{t}_r$ as $\mathbf{X}_{pqr} = \boldsymbol{v}_p \circ \boldsymbol{s}_q \circ \boldsymbol{t}_r$. The $\boldsymbol{v}_p$, $\boldsymbol{s}_q$, and $\boldsymbol{t}_r$ have lengths $|I|$, $|J|$, and $|K|$, respectively.

By combining TD with the rank-one tensor defined above, the approximation of $\mathbf{Y}$ in Equation (1) can be rewritten as a weighted sum of $P \times Q \times R$ rank-one tensors. Here, each weight is an element g_{pqr} of the core tensor $\mathbf{G}$, which quantifies the interaction intensity

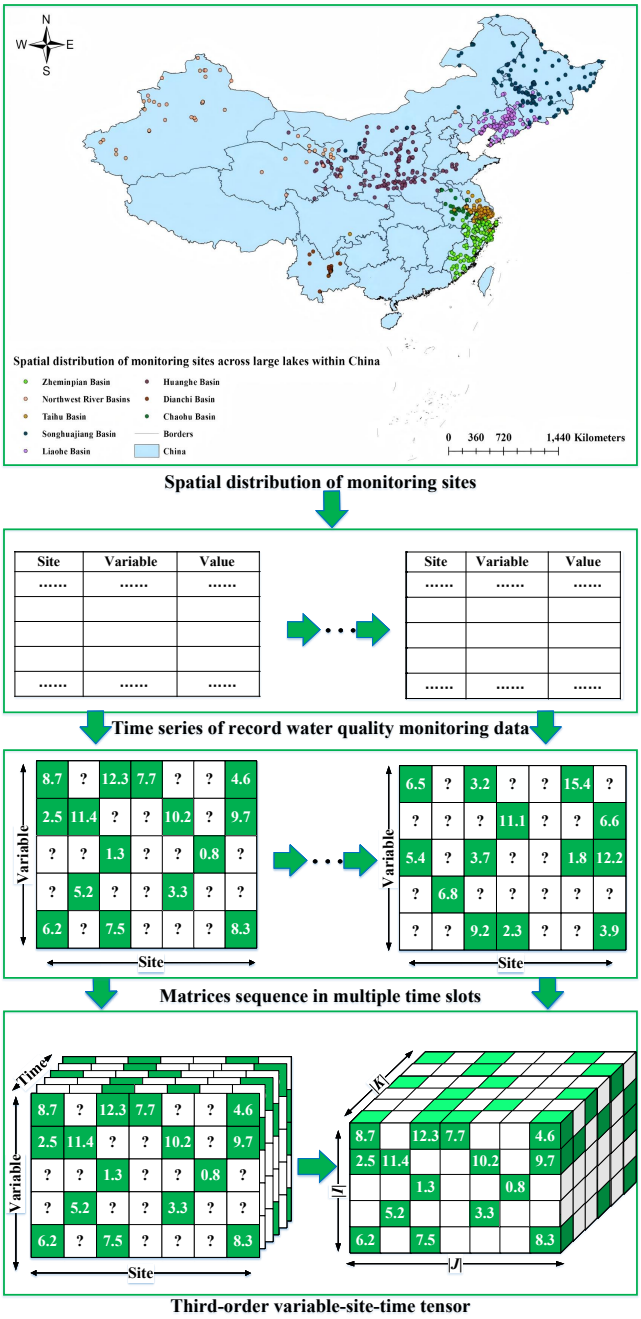


Figure 1. Construction of the variable-site-time HDI tensor

among the corresponding latent features:

$$\mathbf{Y} \approx \hat{\mathbf{Y}} = \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{r=1}^{R} g_{pqr}\,\mathbf{X}_{pqr} = \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{r=1}^{R} g_{pqr}\,\boldsymbol{v}_p \circ \boldsymbol{s}_q \circ \boldsymbol{t}_r. \tag{2}$$

Figure 2 illustrates the construction of a weighted rank-one tensor. The p-th column of V, the q-th column of S, and the r-th column of T are combined to form the pqr-th rank-one tensor, which is then multiplied by the weight g_{pqr}.

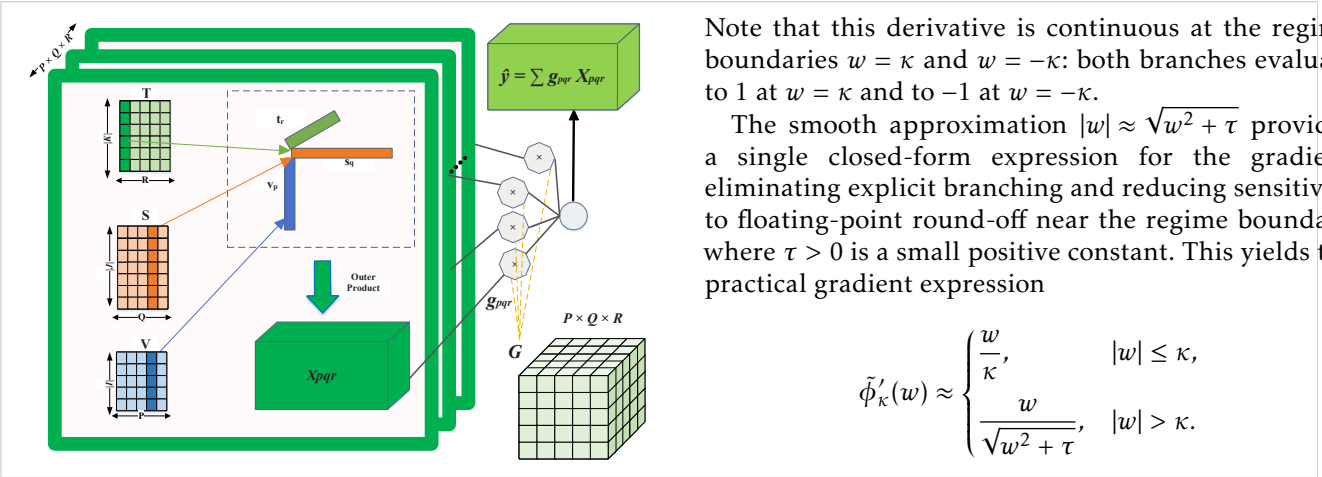


Figure 2. Tucker decomposition approximates a target HDI tensor $\mathbf{Y}$ by a weighted sum of $P \times Q \times R$ rank-one tensors (left). The weight of the pqr-th rank-one tensor is given by the pqr-th element of the core tensor $\mathbf{G}$ (right bottom)

Expanding the outer product in Equation (2) yields an element-wise reconstruction from the core tensor and the corresponding latent factor columns. To estimate the model parameters $\Theta = \{\mathbf{G}, \mathrm{V}, \mathrm{S}, \mathrm{T}\}$, we formulate an objective function that quantifies the reconstruction error between the observed tensor $\mathbf{Y}$ and its approximation on the known entry set Λ. Here, we employ the squared Euclidean distance to construct the loss function:

$$\begin{aligned}\mathcal{E} &= \sum_{(i,j,k)\in\Lambda} \left(y_{ijk} - \hat{y}_{ijk}\right)^2 \\ &= \sum_{(i,j,k)\in\Lambda} \left(y_{ijk} - \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{r=1}^{R} g_{pqr}\, v_{ip}\, s_{jq}\, t_{kr}\right)^2 .\end{aligned} \tag{3}$$

2.3. A Huber Regularization Scheme

The Huber function is a piecewise penalty that behaves quadratically for small arguments and linearly for large arguments [22, 27]. In this work, we adopt a normalized variant in which the linear regime has unit slope. For a scalar w, it is defined as

$$\phi_\kappa(w) = \begin{cases} \dfrac{w^2}{2\kappa}, & |w| \le \kappa, \\ |w| - \dfrac{\kappa}{2}, & |w| > \kappa, \end{cases} \tag{4}$$

where $\kappa > 0$ is a threshold parameter that separates the quadratic and linear regimes. The corresponding derivative is

$$\phi'_\kappa(w) = \begin{cases} \dfrac{w}{\kappa}, & |w| \le \kappa, \\ \operatorname{sign}(w), & |w| > \kappa. \end{cases} \tag{5}$$

Note that this derivative is continuous at the regime boundaries $w = \kappa$ and $w = -\kappa$: both branches evaluate to 1 at $w = \kappa$ and to -1 at $w = -\kappa$.

The smooth approximation $|w| \approx \sqrt{w^2 + \tau}$ provides a single closed-form expression for the gradient, eliminating explicit branching and reducing sensitivity to floating-point round-off near the regime boundary, where $\tau > 0$ is a small positive constant. This yields the practical gradient expression

$$\tilde{\phi}'_\kappa(w) \approx \begin{cases} \dfrac{w}{\kappa}, & |w| \le \kappa, \\ \dfrac{w}{\sqrt{w^2 + \tau}}, & |w| > \kappa. \end{cases} \tag{6}$$

The approximation introduces an perturbation term τ at the regime boundary, which is negligible in practice.

3. An HRL Model

3.1. Objective Function

Directly minimizing the squared error from Equation (3) over the known entry set Λ is ill-posed, since the observed entries are sparse and irregularly distributed, making the solution highly sensitive to the initialization of $\mathbf{G}, \mathrm{V}, \mathrm{S}, \mathrm{T}$. To stabilize the estimation, we incorporate the smoothed Huber penalty $\tilde{\phi}_\kappa$, as defined in Section 2.3, into the objective on a per-observation basis. Specifically, for each observed entry y_{ijk}, the loss comprises the squared reconstruction error plus Huber penalties imposed on every scalar parameter that contributes to the imputation of y_{ijk}. This design, which follows the strategy of [28, 29], weights the This design, which follows the strategy of [28, 29], weights the regularization according to the data density of each parameter and thereby applies stronger regularization to those parameters associated with more observations. The resulting Huber-regularised objective is

$$\begin{aligned}\mathcal{E}_{\text{Huber}} = \sum_{(i,j,k)\in\Lambda} \Bigg(&\Big(y_{ijk} - \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{r=1}^{R} g_{pqr}\, v_{ip}\, s_{jq}\, t_{kr}\Big)^2 \\ &+ \lambda_g \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{r=1}^{R} \tilde{\phi}_\kappa(g_{pqr}) \\ &+ \lambda_f \Big(\sum_{p=1}^{P} \tilde{\phi}_\kappa(v_{ip}) + \sum_{q=1}^{Q} \tilde{\phi}_\kappa(s_{jq}) + \sum_{r=1}^{R} \tilde{\phi}_\kappa(t_{kr}) \Big) \Bigg),\end{aligned} \tag{7}$$

Despite the expressiveness of the Tucker structure, water quality data often exhibit systematic level differences across variables, sites, and time slots. For example, a certain variable may consistently report higher values, or a monitoring station may be affected by local industrial discharge. Such global shifts are difficult to capture with the low-rank interactions alone. To absorb these offsets, we introduce linear bias terms. We introduce bias vectors $\boldsymbol{a} \in \mathbb{R}^{|I|}$, $\boldsymbol{b} \in \mathbb{R}^{|J|}$, and $\boldsymbol{c} \in \mathbb{R}^{|K|}$

for the three modes. The predicted entry becomes

$$\hat{y}_{ijk} = \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{r=1}^{R} g_{pqr}\, v_{ip}\, s_{jq}\, t_{kr} + a_i + b_j + c_k. \tag{8}$$

Substituting Equation (8) into Equation (7) and extending the Huber penalty to the bias parameters, we obtain the complete objective of the HRL:

$$\begin{aligned}\mathcal{E}_{\mathrm{HRL}} = \sum_{(i,j,k)\in\Lambda}\Bigg(&\Big(y_{ijk} - \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{r=1}^{R} g_{pqr}\, v_{ip}\, s_{jq}\, t_{kr} - a_i - b_j - c_k\Big)^2 \\ &+ \lambda_g \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{r=1}^{R} \tilde{\phi}_\kappa(g_{pqr}) \\ &+ \lambda_f\Big(\sum_{p=1}^{P}\tilde{\phi}_\kappa(v_{ip}) + \sum_{q=1}^{Q}\tilde{\phi}_\kappa(s_{jq}) + \sum_{r=1}^{R}\tilde{\phi}_\kappa(t_{kr})\Big) \\ &+ \lambda_b\Big(\tilde{\phi}_\kappa(a_i) + \tilde{\phi}_\kappa(b_j) + \tilde{\phi}_\kappa(c_k)\Big)\Bigg),\end{aligned} \tag{9}$$

where $\lambda_b > 0$ controls the bias regularization. The complete set of model parameters is $\{\mathbf{G}, \mathrm{V}, \mathrm{S}, \mathrm{T}, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}\}$. In practice, the optimisation is carried out via stochastic gradient descent using only the training set Φ, with the gradient of $\tilde{\phi}_\kappa(\cdot)$ given by Equation (6).This training procedure will be detailed in the next section.

3.2. Parameter Learning Scheme

We optimize the objective $\mathcal{E}_{\mathrm{HRL}}$ in Equation (9) via SGD. For a single training sample $(i, j, k) \in \Phi$, denote the residual by

$$e_{ijk} = y_{ijk} - \hat{y}_{ijk}, \tag{10}$$

where $\hat{y}_{ijk}$ is given by (8). The per-sample loss is

$$\begin{aligned}E_{ijk} = &\big(y_{ijk} - \hat{y}_{ijk}\big)^2 \\ &+ \lambda_g \sum_{p,q,r}\tilde{\phi}_\kappa(g_{pqr}) \\ &+ \lambda_f\Big(\sum_{p}\tilde{\phi}_\kappa(v_{ip}) + \sum_{q}\tilde{\phi}_\kappa(s_{jq}) + \sum_{r}\tilde{\phi}_\kappa(t_{kr})\Big) \\ &+ \lambda_b\Big(\tilde{\phi}_\kappa(a_i) + \tilde{\phi}_\kappa(b_j) + \tilde{\phi}_\kappa(c_k)\Big).\end{aligned} \tag{11}$$

We illustrate gradient computation with three representative parameters. For the core tensor element g_{pqr}, the gradient is

$$\frac{\partial E_{ijk}}{\partial g_{pqr}} = -2\, e_{ijk}\, v_{ip}\, s_{jq}\, t_{kr} + \lambda_g\, \tilde{\phi}'_\kappa(g_{pqr}). \tag{12}$$

For the variable-mode latent factor v_{ip} $(p = 1, \ldots, P)$, we have

$$\frac{\partial E_{ijk}}{\partial v_{ip}} = -2\, e_{ijk} \sum_{q=1}^{Q}\sum_{r=1}^{R} g_{pqr}\, s_{jq}\, t_{kr} + \lambda_f\, \tilde{\phi}'_\kappa(v_{ip}). \tag{13}$$

The gradient with respect to the variable bias a_i is

$$\frac{\partial E_{ijk}}{\partial a_i} = -2\, e_{ijk} + \lambda_b\, \tilde{\phi}'_\kappa(a_i). \tag{14}$$

The gradients for s_{jq}, t_{kr}, b_j, and c_k follow the same pattern. As in the standard SGD derivation for factorization models [21, 30], absorbing the constant factor 2 into the learning rate $\eta > 0$, the SGD update for any parameter w reads

$$w \leftarrow w + \eta\, e_{ijk}\, \frac{\partial \hat{y}_{ijk}}{\partial w} - \eta\, \lambda_{(w)}\, \tilde{\phi}'_\kappa(w), \tag{15}$$

where $\lambda_{(w)}$ is the corresponding regularization coefficient.

Substituting the piecewise definition of $\tilde{\phi}'_\kappa$ from Equation (6) and writing the condition with $|w|$ (the smooth approximation $\sqrt{w^2+\tau}$ is implied) yields the following concrete updates.

Core tensor g_{pqr}:

$$g_{pqr} \leftarrow \begin{cases} g_{pqr} + \eta\, e_{ijk}\, v_{ip}\, s_{jq}\, t_{kr} - \dfrac{\eta\lambda_g}{\kappa}\, g_{pqr}, & |g_{pqr}| \le \kappa, \\ g_{pqr} + \eta\, e_{ijk}\, v_{ip}\, s_{jq}\, t_{kr} - \eta\lambda_g\, \dfrac{g_{pqr}}{\sqrt{g_{pqr}^2 + \tau}}, & |g_{pqr}| > \kappa. \end{cases} \tag{16}$$

Variable factor v_{ip}:

$$v_{ip} \leftarrow \begin{cases} v_{ip} + \eta\, e_{ijk} \Sigma_{q,r}\, g_{pqr} s_{jq} t_{kr} - \dfrac{\eta\lambda_f}{\kappa} v_{ip}, & |v_{ip}| \le \kappa, \\ v_{ip} + \eta\, e_{ijk} \Sigma_{q,r}\, g_{pqr} s_{jq} t_{kr} - \eta\lambda_f\, \dfrac{v_{ip}}{\sqrt{v_{ip}^2 + \tau}}, & |v_{ip}| > \kappa. \end{cases} \tag{17}$$

Variable bias a_i:

$$a_i \leftarrow \begin{cases} a_i + \eta\, e_{ijk} - \dfrac{\eta\lambda_b}{\kappa}\, a_i, & |a_i| \le \kappa, \\ a_i + \eta\, e_{ijk} - \eta\lambda_b\, \dfrac{a_i}{\sqrt{a_i^2 + \tau}}, & |a_i| > \kappa. \end{cases} \tag{18}$$

The updates for s_{jq}, t_{kr}, b_j, and c_k follow the same pattern. All parameters are updated sequentially as the SGD routine sweeps through Φ. The learning rate η and Huber constants κ, τ are held fixed, and early stopping on a validation set Ω is employed.

3.3. Algorithm Design and Complexity Analysis

Algorithm 1 outlines the complete training procedure of the HRL model, following the SGD updating rules derived in Section 3.2. For each training sample $(i, j, k) \in \Phi$, the residual e_{ijk} is computed via Equation (8) and Equation (10), after which all involved parameters are updated sequentially: the core tensor $\mathbf{G}$ is updated first, so that the freshly computed values immediately

contribute to the gradient computation for the factor matrices V, S, T; the bias vectors $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$ are updated last. Algorithm 2 for v_{ip}, s_{jq}, t_{kr}, a_i, b_j, and c_k all follow the same Huber-regularized SGD pattern, using their respective gradients and regularization coefficients λ_f and λ_b.

Algorithm 1 HRL

Input: Φ, $|I|$, $|J|$, $|K|$, P, Q, R, η, λ_g, λ_f, λ_b, κ, τ, N

Operation	**Cost**
1. Initialize **G**, V, S, T, $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$.	$\Theta(PQR + \|I\|P + \|J\|Q + \|K\|R)$
2. Initialize $n = 1$.	$\Theta(1)$
while $n \le N$ and not converge **do**	$\times n$
for each $(i, j, k) \in \Phi$ **do**	$\times\|\Phi\|$
3. Compute $\hat{y}_{ijk}$ by (8) and e_{ijk} by (10).	$\Theta(PQR)$
4. Update **G** by (16).	$\Theta(PQR)$
5. Update V, S, T by (17) and its analogs.	$\Theta(PQR)$
6. Update $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$ by (18) and its analogs.	$\Theta(1)$
end for	
7. Evaluate on Ω.	$\Theta(\|\Omega\| \cdot PQR)$
8. $n \leftarrow n + 1$.	$\Theta(1)$
end while	

Output: **G**, V, S, T, $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$

Convergence is monitored on a validation set Ω via RMSE, MAE, and R^2. Training terminates when all three metrics stop improving for n consecutive rounds (early stopping). The best parameters with respect to each metric are retained for final evaluation on the test set Ψ.

The per-round time complexity is $\mathcal{O}(|\Phi| \cdot P \cdot Q \cdot R)$, which is dominated by the triple nested summation in (8) and the updates for the core tensor and factor matrices. The space complexity for storing the model parameters is $\mathcal{O}(|\Phi| + P \cdot Q \cdot R + |I|P + |J|Q + |K|R)$. Additionally, caching the best-performing models with respect to the three evaluation metrics requires $\mathcal{O}\Big((P \cdot Q \cdot R + |I|P + |J|Q + |K|R + |I| + |J| + |K|)\Big)$ storage space. Both time and space complexities scale approximately linearly with the number of training samples $|\Phi|$, demonstrating that HRL is efficient and scalable for large-scale water quality data imputation.

Algorithm 2 Update_G

Input: (i, j, k), e_{ijk}, P, Q, R, η, λ_g, κ, τ, **G**, V, S, T

Operation	**Cost**
for each (p, q, r) **do**	$\times PQR$
$\Delta \leftarrow \eta \cdot e_{ijk} \cdot v_{ip} \cdot s_{jq} \cdot t_{kr}$	
if $\sqrt{g_{pqr}^2 + \tau} \le \kappa$ **then**	
$g_{pqr} \leftarrow g_{pqr} + \Delta - \dfrac{\eta\lambda_g}{\kappa} g_{pqr}$	
else	
$g_{pqr} \leftarrow g_{pqr} + \Delta - \eta\lambda_g \dfrac{g_{pqr}}{\sqrt{g_{pqr}^2 + \tau}}$	
end if	
end for	

Output: **G**

4. Experimental Results and Discussions

4.1. General Settings

Datasets. To evaluate the performance of the proposed HRL model, this paper conducts experiments on eight real-world water quality datasets collected from major watersheds across China [44]. As summarized in Table 2, the datasets encompass monitoring records from diverse hydrological regions, including Dianchi Basin, Chaohu Basin, Northwest River Basins, Songhuajiang Basin, Liaohe Basin, Taihu Basin, Huanghe Basin, and Zheminpian Rivers. Each dataset is organized as a variable–site–time third-order tensor $\mathbf{Y} \in \mathbb{R}^{11 \times |J| \times |K|}$, where the water quality variables are: water temperature (°C), pH (dimensionless), dissolved oxygen (mg/L), conductivity (μS/cm), turbidity (NTU), permanganate index (mg/L), ammonia nitrogen (mg/L), total phosphorus (mg/L), total nitrogen (mg/L), chlorophyll-α (mg/L), and algal density (cells/L). The number of monitoring sites $|J|$ ranges from 30 to 138, and the number of time slots $|K|$ ranges from 8030 to 9954, resulting in tensors of varying scales. The observed data densities range from approximately 22.08% to 39.06%, with an average density of 31.06% across all eight datasets, which is representative of real-world HDI water quality monitoring scenarios.

To create the evaluation protocol, the known entries Λ of each dataset are randomly partitioned into a training set Φ, a validation set Ω, and a test set Ψ with the ratio $\Phi : \Omega : \Psi = 2 : 1 : 7$. Specifically, 20% of the observed entries are used for model training, 10% for early stopping validation, and the remaining 70% for final performance evaluation. This partitioning strategy ensures that the test set is sufficiently large to provide statistically reliable performance estimates, while the validation set is adequate for convergence monitoring. The detailed statistics of the eight experiment datasets are presented in Table 2.

Furthermore, since different water quality variables exhibit vastly different numerical scales (e.g., algal density can reach the order of 10^7 cells/L while pH values lie in a narrow range around 7), a natural logarithm (ln) transformation is applied to all observed entries prior to model training:

$$y_{ijk}^{\text{norm}} = \ln(y_{ijk} + 1), \tag{19}$$

where the additive constant 1 prevents undefined values for zero-valued entries. After imputation, the inverse transformation $\hat{y}_{ijk} = \exp(\hat{y}_{ijk}^{\text{norm}}) - 1$ is applied

to recover the original scale before computing evaluation metrics. This normalization strategy effectively compresses the dynamic range of heterogeneous variables, facilitating stable gradient updates during SGD optimization.

Table 2. Details of the eight real-world water quality datasets

ID	Watershed	No. of Records	$\|I\| : \|J\| : \|K\|$	Density
D1	Dianchi Basin	944,184	11:30:8030	35.63%
D2	Chaohu Basin	1,035,150	11:30:8030	39.06%
D3	Northwest River Basins	1,283,510	11:53:8030	27.42%
D4	Songhuajiang Basin	2,707,289	11:112:9952	22.08%
D5	Liaohe Basin	3,054,901	11:101:9827	27.98%
D6	Taihu Basin	3,378,156	11:105:9461	30.91%
D7	Huanghe Basin	4,309,356	11:138:9954	28.52%
D8	Zheminpian Rivers	4,947,410	11:132:9947	34.25%

Training Settings. The proposed HRL model is trained via SGD with a fixed learning rate $\eta = 5 \times 10^{-4}$ across all datasets. This value is determined through preliminary tuning on representative datasets, where the validation performance remains stable across a wide range of candidate learning rates. Therefore, we adopt this fixed value for all experiments to ensure a fair and consistent comparison. Early stopping is adopted to prevent overfitting: training terminates when none of the validation RMSE, MAE, and R^2 on Ω has improved for $n = 10$ consecutive epochs, or when the maximum number of epochs $N = 1000$ is reached. The model parameters that achieve the best validation RMSE, MAE, and R^2 are saved separately, and each result is used for final evaluation on the test set Ψ.

Evaluation Metrics. The accuracy of missing value imputation directly reflects how well the model captures the latent variable–site–time interactions underlying the water quality dynamics. Hence, this paper focuses on imputation accuracy and adopts three widely-used metrics: root mean square error (RMSE), mean absolute error (MAE), and coefficient of determination (R^2). All three metrics are computed on the test set Ψ as follows:

$$\begin{aligned} \mathrm{RMSE} &= \sqrt{\frac{\sum_{(i,j,k)\in\Psi} \left(y_{ijk} - \hat{y}_{ijk}\right)^2}{|\Psi|}}, \\ \mathrm{MAE} &= \frac{\sum_{(i,j,k)\in\Psi} \left|y_{ijk} - \hat{y}_{ijk}\right|}{|\Psi|}, \\ R^2 &= 1 - \frac{\sum_{(i,j,k)\in\Psi} \left(y_{ijk} - \hat{y}_{ijk}\right)^2}{\sum_{(i,j,k)\in\Psi} \left(y_{ijk} - \bar{y}\right)^2}, \end{aligned} \tag{20}$$

where y_{ijk} and $\hat{y}_{ijk}$ denote the ground-truth and predicted values for the (i, j, k)-th entry in the test set, respectively, $|\Psi|$ is the cardinality of Ψ, and $\bar{y}$ is the global average of all observed values in Ψ.

4.2. Model Comparison

First, the performance of our proposed HRL model is compared with state-of-the-art models that are capable of recovering missing entries in HDI water quality tensors. The compared models are summarized as follows.

(a) **M1:** A high-dimension-oriented imputation model [31]. It handles multi-dimensional data using tensor concepts and imputes missing values via CP decomposition together with a reconstruction optimization algorithm.

(b) **M2:** A tensor completion algorithm-based model [32]. It employs CP decomposition and a gradient descent-based mechanism to assess performance by imputing the missing entries in an incomplete tensor.

(c) **M3:** A sparse and graph-regularized canonical polyadic tensor factorization model [33], which incorporates graph Laplacian regularization to preserve the local geometric structure among entities while factorizing the tensor.

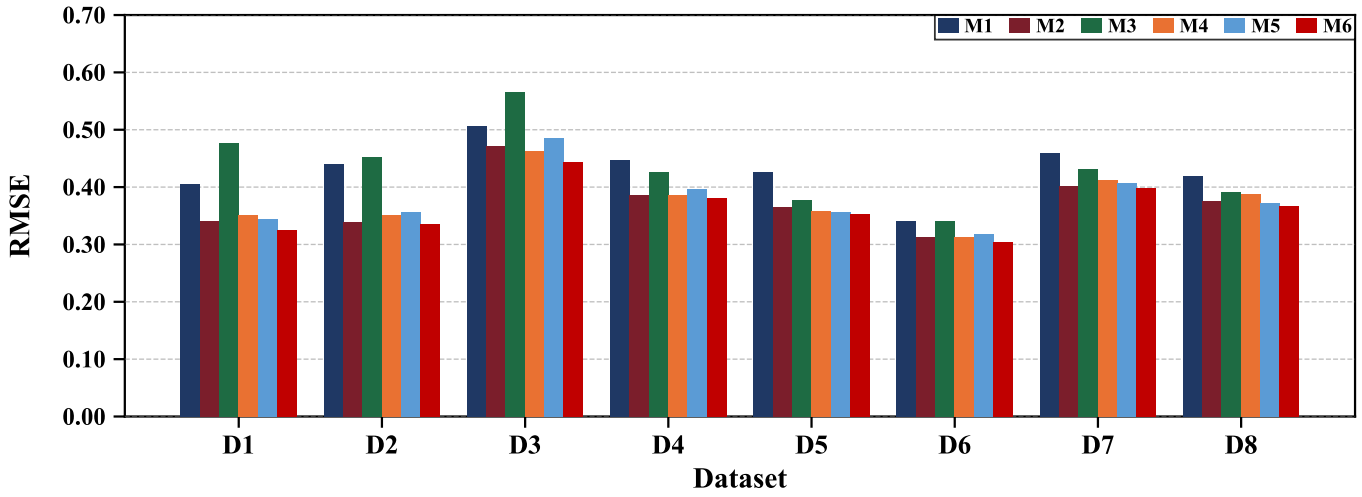
(d) **M4:** A fine-grained regularized nonnegative latent factorization of tensors model [34], abbreviated as FRNL (**F**ine-grained **R**egularized **N**onnegative **L**FT). It imposes fine-grained regularization on the factor matrices under the nonnegativity constraint.

(e) **M5:** An adaptively bias-extended nonnegative LFT model [45], which augments the nonnegative CP factorization with adaptive bias terms to absorb systematic level shifts across modes.

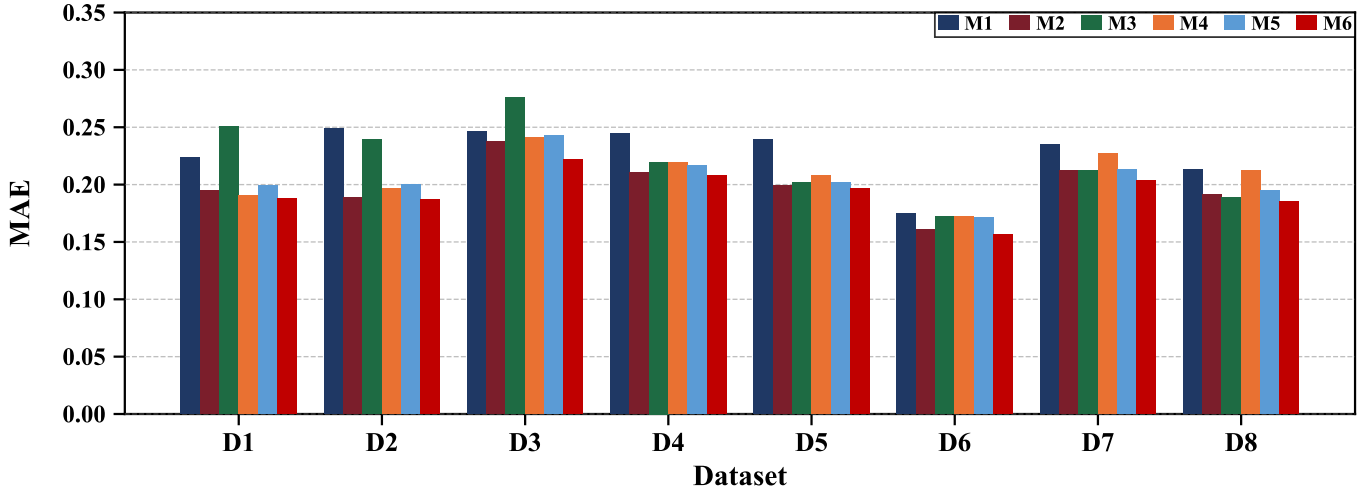
(f) **M6:** The proposed HRL model in this study, as detailed in Section 3.

To ensure a fair comparison, the latent feature ranks for all modes are set identically for all models, guaranteeing that all models operate under the same representational capacity. The remaining hyperparameters of each competitor are individually tuned to attain their best achievable performance. Every model is independently executed for 20 rounds on each dataset, and the standard deviations are included in the experiments. The RMSE and MAE comparisons over D1–D8 are visualized in Figure 3 and Table 3, while the corresponding training time is summarized in Table 4. From these results, this paper draws the following findings.

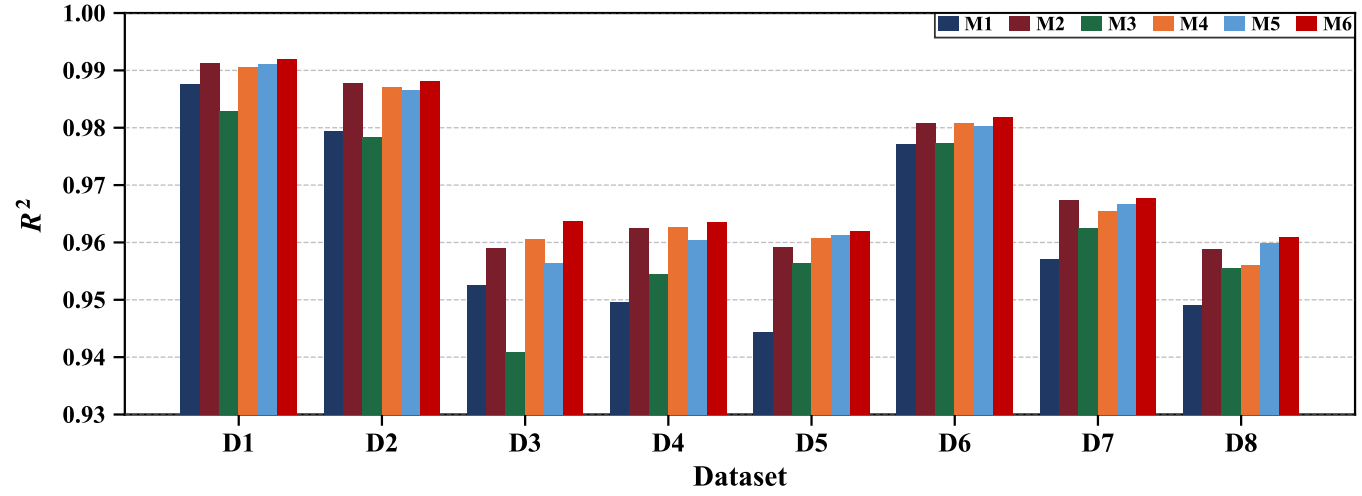
(a) HRL achieves the highest imputation accuracy across all datasets. As shown in Table 3,HRL



(a) RMSE comparison of M1–M6 on D1–D8.



(b) MAE comparison of M1–M6 on D1–D8.



(c) R^2 comparison of M1–M6 on D1–D8.

Figure 3. Imputation accuracy comparison across the eight real-world water quality datasets. Lower bars indicate better performance for RMSE and MAE, while higher bars indicate better performance for R^2. The proposed HRL (M6) achieves the best results on all datasets across all three metrics

Table 3. RMSE, MAE and R^2 (mean±std over 20 rounds) of the compared models on D1–D8. The best result in each row is highlighted in **bold**

Dataset	Metric	M1	M2	M3	M4	M5	M6
D1	RMSE	0.4048±0.0220	0.3406±0.0034	0.4772±0.0050	0.3516±0.0020	0.3436±0.0077	**0.3258±0.0003**
	MAE	0.2237±0.0083	0.1946±0.0015	0.2507±0.0030	0.1903±0.0014	0.1992±0.0038	**0.1878±0.0008**
	R^2	0.9875±0.0014	0.9912±0.0002	0.9828±0.0014	0.9906±0.0010	0.9910±0.0014	**0.9919±0.0003**
D2	RMSE	0.4409±0.0240	0.3399±0.0011	0.4526±0.0134	0.3512±0.0037	0.3564±0.0056	**0.3353±0.0020**
	MAE	0.2490±0.0117	0.1890±0.0011	0.2396±0.0043	0.1966±0.0015	0.2003±0.0022	**0.1873±0.0016**
	R^2	0.9794±0.0022	0.9878±0.0001	0.9784±0.0023	0.9870±0.0003	0.9866±0.0014	**0.9881±0.0001**
D3	RMSE	0.5069±0.0059	0.4711±0.0062	0.5666±0.0148	0.4626±0.0052	0.4864±0.0090	**0.4438±0.0009**
	MAE	0.2467±0.0068	0.2378±0.0043	0.2756±0.0052	0.2407±0.0022	0.2432±0.0050	**0.2216±0.0004**
	R^2	0.9525±0.0015	0.9590±0.0011	0.9408±0.0041	0.9605±0.0011	0.9564±0.0017	**0.9637±0.0001**
D4	RMSE	0.4476±0.0143	0.3865±0.0030	0.4261±0.0187	0.3857±0.0038	0.3971±0.0038	**0.3806±0.0023**
	MAE	0.2446±0.0147	0.2109±0.0019	0.2197±0.0053	0.2190±0.0012	0.2166±0.0026	**0.2077±0.0014**
	R^2	0.9496±0.0032	0.9624±0.0006	0.9544±0.0041	0.9626±0.0004	0.9604±0.0018	**0.9636±0.0004**
D5	RMSE	0.4263±0.0187	0.3655±0.0030	0.3783±0.0033	0.3586±0.0027	0.3560±0.0039	**0.3529±0.0014**
	MAE	0.2392±0.0113	0.1988±0.0013	0.2014±0.0020	0.2078±0.0015	0.2022±0.0027	**0.1967±0.0013**
	R^2	0.9443±0.0049	0.9591±0.0006	0.9563±0.0017	0.9607±0.0003	0.9612±0.0012	**0.9619±0.0003**
D6	RMSE	0.3408±0.0101	0.3132±0.0020	0.3403±0.0040	0.3128±0.0021	0.3175±0.0037	**0.3040±0.0013**
	MAE	0.1751±0.0038	0.1610±0.0006	0.1724±0.0024	0.1724±0.0014	0.1715±0.0026	**0.1564±0.0008**
	R^2	0.9772±0.0038	0.9807±0.0006	0.9773±0.0015	0.9808±0.0002	0.9802±0.0014	**0.9818±0.0002**
D7	RMSE	0.4601±0.0188	0.4020±0.0019	0.4307±0.0031	0.4130±0.0026	0.4063±0.0039	**0.3992±0.0017**
	MAE	0.2348±0.0191	0.2121±0.0019	0.2122±0.0023	0.2275±0.0017	0.2134±0.0055	**0.2033±0.0007**
	R^2	0.9571±0.0035	0.9673±0.0003	0.9625±0.0015	0.9655±0.0007	0.9666±0.0016	**0.9677±0.0003**
D8	RMSE	0.4185±0.0054	0.3760±0.0017	0.3915±0.0026	0.3885±0.0028	0.3715±0.0037	**0.3669±0.0014**
	MAE	0.2134±0.0085	0.1911±0.0011	0.1887±0.0019	0.2123±0.0017	0.1947±0.0029	**0.1855±0.0008**
	R^2	0.9491±0.0013	0.9589±0.0004	0.9555±0.0022	0.9561±0.0003	0.9599±0.0016	**0.9609±0.0003**

Table 4. Training time (seconds, mean±std over 20 rounds) of the compared models on D1–D8. The best result in each row is highlighted in **bold**

Dataset	Metric	M1	M2	M3	M4	M5	M6
D1	RMSE-Time	261±34	208±21	54±8	1048±67	323±36	**46±9**
	MAE-Time	260±35	210±21	56±10	1067±71	322±35	**47±7**
	R^2-Time	255±47	173±21	53±8	929±99	293±48	**38±6**
D2	RMSE-Time	260±49	200±17	102±20	1044±59	328±58	**40±11**
	MAE-Time	259±48	235±20	114±25	1052±68	329±55	**76±5**
	R^2-Time	253±44	179±20	101±21	937±90	281±78	**36±5**
D3	RMSE-Time	380±65	208±8	126±41	1211±76	365±90	**43±5**
	MAE-Time	385±64	248±13	127±42	1201±76	380±89	**45±4**
	R^2-Time	374±65	201±12	126±41	1134±108	358±97	**41±4**
D4	RMSE-Time	709±172	437±27	191±21	2641±357	716±223	**74±15**
	MAE-Time	710±170	434±26	194±28	2637±343	809±157	**98±12**
	R^2-Time	407±82	269±15	134±35	1531±95	335±78	**71±12**
D5	RMSE-Time	688±31	602±83	220±26	2380±66	899±183	**156±14**
	MAE-Time	684±31	604±78	231±28	2371±62	872±194	**182±6**
	R^2-Time	704±170	419±27	191±21	2559±328	641±195	**141±13**
D6	RMSE-Time	626±72	551±92	254±26	2679±140	915±269	**233±29**
	MAE-Time	626±72	548±92	267±42	2655±136	1015±171	**238±11**
	R^2-Time	681±32	588±74	219±26	2353±37	827±196	**218±15**
D7	RMSE-Time	854±72	479±39	280±18	3739±205	1322±477	**260±28**
	MAE-Time	860±70	479±48	310±18	3758±220	1331±463	**255±47**
	R^2-Time	616±69	514±210	253±26	2478±204	824±267	**251±30**
D8	RMSE-Time	976±148	691±133	418±22	3827±351	1497±180	**193±25**
	MAE-Time	971±152	718±106	437±10	3842±373	1507±186	**241±10**
	R^2-Time	852±70	460±41	279±18	3567±270	1251±432	**189±8**

obtains the lowest RMSE and MAE on every one of the eight datasets, i.e., 8/8 wins on RMSE and 8/8 on MAE, totaling 16/16 evaluation metrics. Taking D1 as an example, the RMSE of HRL is 0.3258, which represents improvements of 19.52%, 4.35%, 31.73%, 7.34%, and 5.18% over M1, M2, M3, M4, and M5, respectively; the corresponding MAE improvements against M1–M5 are 16.05%, 3.49%, 25.09%, 1.31%, and 5.72%, respectively. On the sparsest dataset D4 (density 22.08%), HRL's RMSE (0.3806) and MAE (0.2077) show improvements of 14.97% and 15.09% over M1. Furthermore, these improvements are complemented by consistently superior R^2 values. For D1, HRL yields an R^2 of 0.9919, outperforming all five competing models; for the sparsest D4, it attains 0.9636, surpassing M2 (0.9624) and M4 (0.9626). A higher R^2 indicates that HRL explains a larger share of the observed variance in water quality dynamics, suggesting its potential to support more reliable trend identification and anomaly warning in water resource management. Overall, the improvements across RMSE, MAE, and R^2 confirm that the Huber-regularized Tucker structure remains highly effective under extreme sparsity.

(b) HRL exhibits markedly stronger robustness, reflected by its minimal performance variance. The standard deviations in Table 3 further reveal that HRL is the most stable model. On D1, the RMSE standard deviation of HRL is merely 0.0003, while those of M1, M3, and M5 are 0.0220, 0.0050, and 0.0077—i.e., 73×, 16×, and 25× larger, respectively. Similar stability advantages persist on D2, D7, and D8. This robustness stems from the smoothed Huber penalty $\tilde{\phi}_\kappa(\cdot)$: it behaves quadratically for small residuals to maintain smooth convergence, and linearly for large residuals, thereby suppressing the influence of outliers and sensor anomalies that commonly occur in water quality measurements and would otherwise dominate the objective function.

(c) HRL delivers the best computational efficiency among all compared models. Table 4 shows that HRL is the fastest model on every dataset, with training time ranging from 40 to 260 seconds and scaling linearly with the number of observed entries. Notably, it is consistently faster than M3, which exhibits the worst accuracy among the five competitors (Table 3), highlighting an unfavorable accuracy–efficiency trade-off for that lightweight design. Meanwhile, M4, the model whose accuracy is closest to HRL, requires 1048–3842 seconds on the large datasets D4–D8, which is 11 to 27 times longer than HRL's training time. Thus, HRL simultaneously realizes the highest accuracy and the lowest computational cost, corroborating the linear-complexity analysis in Section 3.3 and confirming its practicality for large-scale water quality data imputation.

(d) The performance gain of the proposed HRL model is statistically significant across our experiments. To further investigate these results, the Friedman test [43] is employed to verify their statistical significance, as reported in Table 3 and Table 4. The mean rank of each model's efficiency and accuracy is presented in Table 5. Let $r_{d,m}$ denote the rank of the m-th model among M compared models on the d-th of D test cases. The test computes the average rank of each model across all datasets as $A_m = \sum_{d=1}^{D} r_{d,m}/D$. Accordingly, we obtain the following Friedman statistic:

$$\chi_F^2 = \frac{12D}{M(M+1)}\left[\sum_m A_m^2 - \frac{M(M+1)^2}{4}\right]. \quad (21)$$

The testing score is calculated as:

$$F_F = \frac{(D-1)\chi_F^2}{D(M-1)-\chi_F^2}. \quad (22)$$

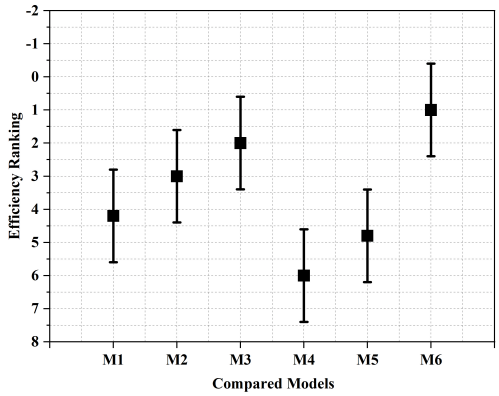
Note that the testing score is distributed based on the F-distribution with $M-1$ and $(M-1)(D-1)$ degrees of freedom. Therefore, if F_F is better than the corresponding critical value, we reject all assumptions of model equivalence with critical level α.

Our experiments compared six models on eight test cases. Accordingly, the constants are $M = 6$, $D = 8$, and F_F follows an F-distribution with degrees of freedom 5 and 35. At $\alpha = 0.05$, the critical value for this configuration is 2.293, so the null hypothesis is rejected whenever the test statistic exceeds this threshold. As reported in Table 3 and Table 4, the test statistics equal 55.15 for accuracy and 1185.0 for efficiency. We therefore conclude that M1–M6 exhibit statistically significant performance differences at the 95% confidence level. Table 5 clearly indicates that M6 attains the best average rank, demonstrating its superiority over the competing models in both imputation accuracy and computational efficiency.

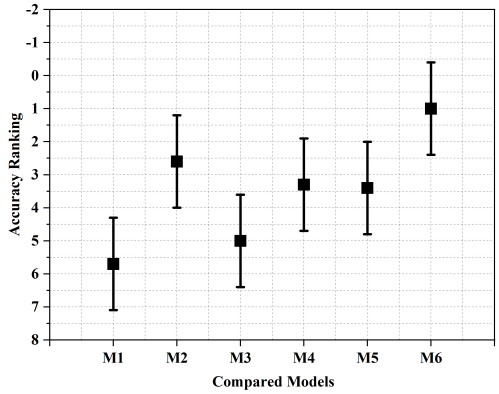
Table 5. Average ranks of all compared models

Rank	M1	M2	M3	M4	M5	M6
Efficiency	4.2	3.0	2.0	6.0	4.8	**1.0**
Accuracy	5.7	2.6	5.0	3.3	3.4	**1.0**

Furthermore, the Nemenyi test [43] is then performed to determine whether M6, the proposed HRL model,



(a) Efficiency



(b) Accuracy

Figure 4. Nemenyi test results comparing computational efficiency and imputation accuracy

surpasses its peers in both computational efficiency and imputation accuracy within the experiments. According to the Nemenyi test, two models are significantly different if their rank difference exceeds the critical value. The critical value is calculated as [43]:

$$\delta = q_\alpha \sqrt{\frac{M(M+1)}{6D}}, \tag{23}$$

where the constant q_α is decided by the studentized range statistics[43] . With six compared models on eight testing cases and $\alpha = 0.1$[43] , the critical difference is calculated as $\delta = 1.398$. This indicates that two models with a rank difference higher than 1.398 have significant difference in performance with a confidence level at 90%.

From the Nemenyi analysis results shown in Figure 4, we conclude that M6 significantly surpasses M1, M2, M4, and M5 in computational efficiency, and significantly exceeds M1, M3, M4, and M5 in imputation accuracy for missing entries of water quality tensors. For the remaining models, although the performance differences do not reach the threshold of statistical significance, M6 still maintains a clear efficiency and accuracy advantage over them across all testing cases. Consequently, the proposed HRL model simultaneously achieves high computational speed, high predictive accuracy for missing water quality data, and strong stability.

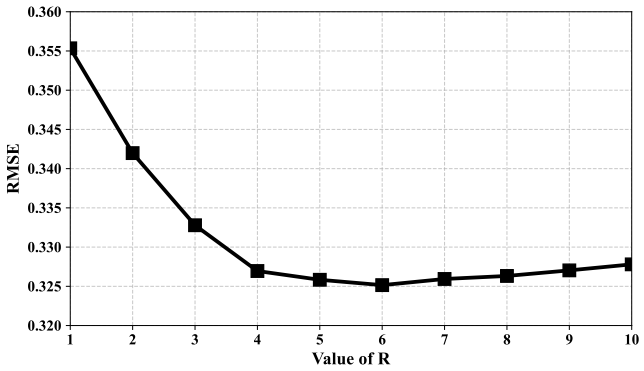
In summary, the empirical comparison demonstrates that the proposed HRL model advances imputation accuracy, robustness to outliers and initialization, and computational efficiency all at once, outperforming five representative SOTA models on all eight real-world water quality datasets.

4.3. Effects of Rank R, Regularization Coefficient λ, and Huber Threshold κ

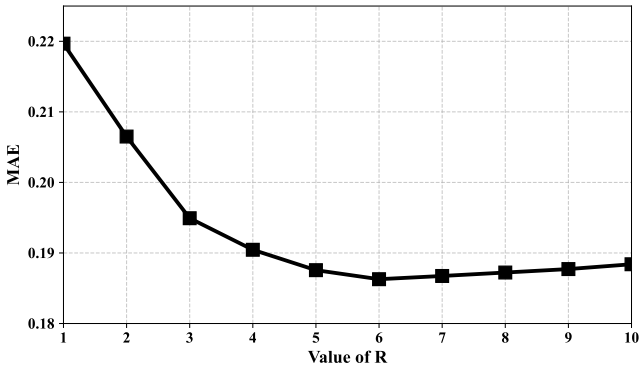
The performance of HRL depends on three key hyperparameters: the latent factor dimension R, where we set $P = Q = R$ for simplicity; the regularization coefficient λ, with all components sharing a single value; and the Huber threshold κ that separates the quadratic and linear regimes of the penalty. We study their impact on dataset D1.

We vary $R \in \{1, 2, \ldots, 10\}$ while keeping all other settings fixed. Figure 5 reports the test RMSE in the upper panel and the test MAE in the lower panel. The R^2 metric exhibits a consistent trend with these two indicators, yet with a notably smaller magnitude of variation, so its curve is omitted for brevity. Accuracy clearly depends on R but saturates rapidly. For small R both errors drop sharply: the test RMSE decreases from 0.3553 at $R = 1$ to 0.3258 at $R = 5$, reaching its minimum 0.3252 at $R = 6$, which correspond to improvements of 8.30% and 8.47% over $R = 1$; the MAE falls from 0.2197 to 0.1876 at $R = 5$ and 0.1863 at $R = 6$, yielding reductions of 14.61% and 15.20%. For $R > 6$ the errors plateau and then slowly increase: at $R = 9$ the RMSE rises back to 0.3270 and the MAE to 0.1877, representing degradations of 0.55% and 0.75% relative to $R = 6$, a clear sign of over-parameterization. Crucially, the marginal gain from $R = 5$ to $R = 6$ is merely 0.18% in RMSE and 0.70% in MAE, where the training time grows from 56.7 s to 82.2 s; from $R = 1$ to $R = 10$ the time surges from 3.2 s to 196.4 s, consistent with the linear-in-R complexity in Section 3.3. Balancing accuracy and efficiency, we use $R = 5$ for the smaller datasets D1–D3 and $R = 6$ for the larger D4–D8, matching the saturation point in Figure 5.

We evaluate λ on a log-uniform grid $\{10^{-6}, 3\times10^{-6}, 10^{-5}, 3\times10^{-5}, 10^{-4}, 3\times10^{-4}\}$ and κ on $\{0.01, 0.03, 0.1, 0.3, 1, 3\}$, yielding 36 combinations. The resulting test RMSE and MAE are shown as contour maps in Figure 6. The model exhibits remarkable insensitivity to these two hyperparameters: across the entire grid, RMSE varies only from 0.3248 to 0.3310, a

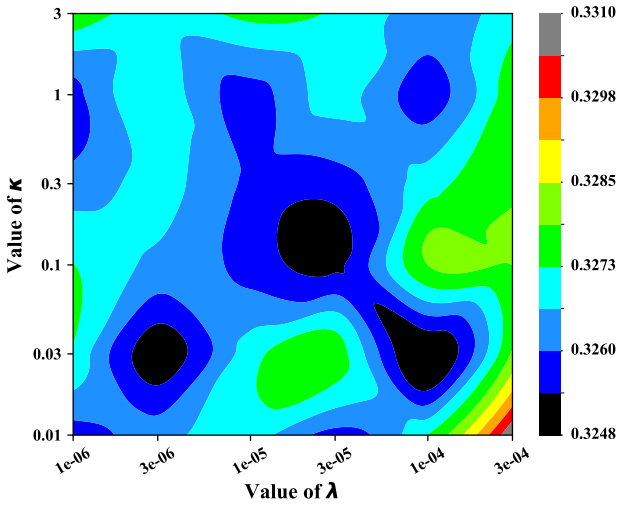


(a) RMSE on D1

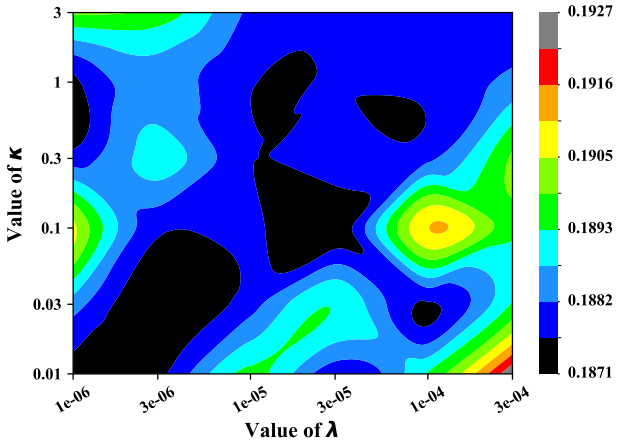


(b) MAE on D1

Figure 5. Impact of the LF dimension R on D1



(a) RMSE



(b) MAE

Figure 6. Sensitivity to λ (horizontal, log scale) and κ (vertical, log scale) on D1. Darker colors indicate lower error

1.91% spread, and MAE from 0.1871 to 0.1927, a 2.99% spread. A wide dark plateau in both panels indicates that almost any moderate choice yields near-optimal performance, a direct benefit of the smoothed Huber penalty which avoids the sharp sensitivity of a pure L_1 or L_2. The best results concentrate in a moderate region: the lowest RMSE, 0.3248, is attained at $\lambda = 10^{-4}$ and $\kappa = 0.03$, while the lowest MAE, 0.1871, is attained at $\lambda = 3 \times 10^{-6}$ and $\kappa = 0.03$. These two optimal configurations are practically interchangeable: the RMSE-optimal point gives an MAE of 0.1876, only 0.27% above the MAE-optimal value, and the MAE-optimal point yields an RMSE of 0.3251, only 0.09% above the RMSE-optimal value. Only the extreme right edge, where $\lambda = 3 \times 10^{-4}$, causes noticeable degradation, with RMSE rising to 0.3310 and MAE to 0.1927, confirming that over-regularization must be avoided; otherwise a coarse empirical tuning on a small data subset is sufficient.

4.4. Summary

Based on the empirical results, we summarize that HRL is equipped with high imputation accuracy and highly competitive computational efficiency. It normally outperforms its peers in terms of overall performance. Therefore, the proposed HRL model is an effective and robust imputation model for HDI water quality tensors.

5. Conclusion and Future Work

This study presents an HRL model based on the Tucker decomposition framework. By incorporating a Huber regularization scheme into the learning objective, the proposed HRL model effectively enhances robustness against sensor outliers while preserving the smoothness of LF learning. To the best of our knowledge, this is the first attempt to introduce Huber regularization into LFT-TD for missing data imputation. Extensive experiments demonstrate that the proposed HRL model consistently achieves superior imputation accuracy and higher computational efficiency th an state-of-the-art models. Therefore, HRL offers an effective and robust solution for reconstructing large- scale incomplete water

quality data, thereby supporting reliable environmental analysis and informed decision-making.

However, the addition of the core tensor increases the number of parameters. In addition, the HRL model may struggle in extremely sparse settings where observations are insufficient to reliably estimate the core tensor, and it does not explicitly handle multimodal missing patterns such as structured block-wise missingness. To address this, future work can adopt model pruning [35, 36] or explore other tensor decomposition schemes such as tensor ring decomposition [37] to reduce complexity. Moreover, incorporating adaptive optimizers [38, 39] and hyperparameter tuning techniques [40, 41] will help avoid laborious manual tuning and further accelerate training.

Acknowledgment

This study was supported by the open project of Dazhou Key Laboratory of Government Data Security (ZSAQ202508), Research Start-up Fund for High-Level Talents of Xihua University (RX2600000693), the open project of Dazhou Key Laboratory of Police Intelligent Robot (JYZJ202602), the Open Fund of Sichuan Provincial University Key Laboratory of Intelligent Unmanned System Technology (No.ZWXJ202612), and the Natural Science Foundation of Xinjiang Uygur Autonomous Region (No. 2024D01A141).

References

[1] X. Wu, L. Wang, M. Ge, J. Jiang, Y. Cai, and B. Yang, "Adaptive biases-incorporated latent factorization of tensors for predicting missing data in water quality monitoring networks," *Front. Phys.* 2025;13:1587012.

[2] X. Wu, L. Wang, M. Xie, and K. Shan, "A well-designed regularization scheme for latent factorization of high-dimensional and incomplete water-quality tensors from sensor networks," in *Proc. 19th IEEE Int. Conf. Mobility, Sensing and Networking*. 2023. p. 596–603.

[3] R. Marcé, G. George, P. Buscarinu, M. Deidda, J. Dunalska, E. de Eyto, G. Flaim, H.-P. Grossart, V. Istvanovics, M. Lenhardt, E. Moreno-Ostos, B. Obrador, I. Ostrovsky, D. C. Pierson, J. Potužák, S. Poikane, K. Rinke, S. Rodríguez-Mozaz, P. A. Staehr, K. Šumberová, G. Waajen, G. A. Weyhenmeyer, K. C. Weathers, M. Zion, B. W. Ibelings, and E. Jennings, "Automatic high frequency monitoring for improved lake and reservoir management," *Environ. Sci. Technol.* 2016;50(20):10780–10794.

[4] J. He, Z. Wang, W. Huang, F. Zhou, A. Wang, X. Wu, P. Chen, T. He, J. Gan, P. Wei, Z. Li, and C. Wu, "An efficient hybrid model with Harris Hawks optimization algorithm for predicting oat water," *EAI Endorsed Trans. AI and Robotics* 2026;5.

[5] A. Al Mamun, M. F. S. Tanoor, A. K. M. Masum, T. Bhuiyan, and M. M. Hassan, "Explainable AI for customer churn prediction in the energy sector using ensemble machine learning models," *EAI Endorsed Trans. AI and Robotics* 2026;5.

[6] O. Dagtekin and N. Dethlefs, "Imputation of partially observed water quality data using self-attention LSTM," in *Proc. IEEE Int. Joint Conf. Neural Networks (IJCNN)*. 2022. p. 1–8.

[7] J. Choi, K. J. Lim, and B. J. Ji, "Robust imputation method with context-aware voting ensemble model for management of water-quality data," *Water Res.* 2023;243:120369.

[8] Y. Zhang and P. Thorburn, "Handling missing data in near real-time environmental monitoring: a system and a review of selected methods," *Future Gener. Comput. Syst.* 2022;128:63–72.

[9] S. Ouyang, "Evaluation of river water quality monitoring stations by principal component analysis," *Water Res.* 2005;39(12):2621–2635.

[10] X. Wu, K. Shan, L. Wang, J. Wang, and M. Shang, "Spatiotemporal water quality data reconstruction: A tensor factorization framework," *Ecol. Inform.* 2025;90:103283.

[11] E. E. Papalexakis, C. Faloutsos, and N. D. Sidiropoulos, "Tensors for data mining and data fusion: Models, applications, and scalable algorithms," *ACM Trans. Intell. Syst. Technol.* 2017;8(2):art. 16, pp. 1–44.

[12] X. Luo, M. Zhou, S. Li, Z. You, Y. Xia, and Z. Feng, "A nonnegative latent factor model for large-scale sparse matrices in recommender systems and its application," *IEEE Trans. Neural Netw. Learn. Syst.* 2016;27(3):579–592.

[13] Y. Ji, Q. Wang, X. Li, and J. Liu, "A survey on tensor techniques and applications in machine learning," *IEEE Access* 2019;7:162950–162990.

[14] Y. Wu, H. Tan, Y. Li, J. Zhang, and X. Chen, "A fused CP factorization method for incomplete tensors," *IEEE Trans. Neural Netw. Learn. Syst.* 2019;30(3):751–764.

[15] W. Zhang, H. Sun, X. Liu, and X. Guo, "Temporal QoS-aware Web service recommendation via non-negative tensor factorization," in *Proc. 23rd Int. Conf. World Wide Web*. 2014. p. 585–596.

[16] X. Wu, K. Shan, F. Recknagel, L. Wang, and M. Shang, "Enhanced tensor factorization for spatiotemporal imputation of high-frequency water quality monitoring data," *Environ. Model. Softw.* 2025;193:Art. no. 106667.

[17] E. Acar, D. Dunlavy, T. Kolda, and M. Morup, "Scalable tensor factorizations with missing data," in *Proc. SIAM Int. Conf. Data Mining*. 2010. p. 701–712.

[18] Q. L. Liu, L. Z. Lu, and Z. Chen, "Non-negative Tucker decomposition with graph regularization and smooth constraint for clustering," *Pattern Recognit.* 2023;148:110207.

[19] S. Fang, A. Narayan, R. M. Kirby, and S. Zhe, "Bayesian continuous-time Tucker decomposition," in *Proc. 39th Int. Conf. Mach. Learn. (ICML)*. PMLR; 2022. p. 6235–6245.

[20] Y. Zhang, Y. Liu, and Y. Wang, "Displacement data imputation in urban Internet of Things system based on Tucker decomposition with L2 regularization," *IEEE Internet Things J.* 2022;9(18):13315–13326.

[21] P. Tang, T. Ruan, H. Wu, and X. Luo, "Temporal pattern-aware QoS prediction by Biased Non-negative Tucker Factorization of tensors," *Neurocomputing* 2024;582:127447.

[22] P. J. Huber, "Robust estimation of a location parameter," *Ann. Math. Stat.* 1964;35(1):73–101.

[23] P. Ochs, A. Dosovitskiy, T. Brox, and T. Pock, "On iteratively reweighted algorithms for nonsmooth

nonconvex optimization in computer vision," *SIAM J. Imaging Sci.* 2015;8(1):331–372.
[24] L. R. Tucker, "Implications of factor analysis of three-way matrices for measurement of change," in *Problems in Measuring Change*, C. W. Harris, editor. Madison, WI, USA: Univ. Wis. Press; 1963. p. 122–137.
[25] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika* 1966;31(3):279–311.
[26] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Trans. Signal Process.* 2017;65(13):3551–3582.
[27] P. J. Huber and E. M. Ronchetti, *Robust Statistics*. 2nd ed. Hoboken, NJ, USA: Wiley; 2009.
[28] Y. Koren and R. Bell, "Advances in collaborative filtering," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, and B. Shapira, editors. Boston, MA, USA: Springer; 2015. p. 77–118.
[29] A. Paterek, "Improving regularized singular value decomposition for collaborative filtering," in *Proc. KDD Cup Workshop*. 2007. p. 39–42.
[30] R. Gemulla, E. Nijkamp, P. J. Haas, and Y. Sismanis, "Large-scale matrix factorization with distributed stochastic gradient descent," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*. 2011. p. 69–77.
[31] S. Wang, Y. Ma, B. Cheng, F. Yang, and R. Chang, "Multi-dimensional QoS prediction for service recommendations," *IEEE Trans. Serv. Comput.* 2019;12(1):47–57.
[32] X. Su, M. Zhang, Y. Liang, Z. Cai, L. Guo, and Z. Ding, "A tensor-based approach for the QoS evaluation in service-oriented environments," *IEEE Trans. Netw. Serv. Manag.* 2021;18(3):3843–3857.
[33] H. Che, B. Pan, M.-F. Leung, Y. Cao, and Z. Yan, "Tensor factorization with sparse and graph regularization for fake news detection on social networks," *IEEE Trans. Comput. Soc. Syst.* 2024;11(4):4888–4898.
[34] H. Wu, Y. Qiao, and X. Luo, "A fine-grained regularization scheme for nonnegative latent factorization of high-dimensional and incomplete tensors," *IEEE Trans. Serv. Comput.* 2024;17(6):3006–3021.
[35] V.-K. Pham, M.-D. Do, T.-N. Dang, and L. Tran, "Hardware-aware INT8 quantization and FPGA deployment of MobileNetV2 for real-time facial landmark detection," *EAI Endorsed Trans. AI and Robotics* 2026.
[36] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," in *Proc. 4th Int. Conf. Learn. Represent. (ICLR)*. 2016.
[37] K. K. M. Mickelin and S. Wahls, "Tensor Ring Decomposition for Efficient Subspace Projections," *SIAM J. Matrix Anal. Appl.* 2022;43(2):720–743.
[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*. 2015.
[39] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.* 2011;12:2121–2159.
[40] K. Kalaivani, P. Deepan, G. Ganesh, J. Ravichandran, and S. Dhiravidaselvi, "Hyperband-optimized convolutional neural network model for efficient brain tumor classification and prediction," *EAI Endorsed Trans. AI and Robotics* 2025.
[41] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning hyperparameters," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*. 2012. p. 2951–2959.
[42] L. Wang, X. Wu, Y. Yu, H. Liu, K. Shan, and B. Yang, "Nonnegative latent factorization of tensors with swish regularization for water quality data imputation," in *Proc. 2024 12th Int. Conf. Commun. Broadband Netw. (ICCBN)*. 2024. p. 201–207.
[43] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.* 2006;7:1–30.
[44] X. Wu, J. Hu, Y. Wang, and H. Tu, "AHFM water quality datasets collected from thirteen typical watersheds across China [Data set]," figshare; 2026. [Online]. Available: https://doi.org/10.6084/m9.figshare.32527353
[45] X. Xu, M. Lin, X. Luo, and Z. Xu, "An adaptive bias-extended non-negative latent factorization of tensors model for accurately representing the dynamic QoS data," *IEEE Trans. Serv. Comput.* 2025;18(2):603–617.
[46] A. Elragig, A. J. Moshayedi, D. Bassir, Z. Khan, A. Kolahdooz, and A. S. Khan, "Smart agro-ecosystem: A review of LLM-based robotic systems for sensing and decision support in precision agriculture," *EAI Endorsed Trans. AI and Robotics* 2026;5.