KDD2024 BARCELONA, SPAIN

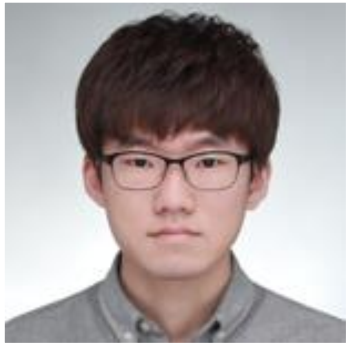
KAIST

COMPUTER SCIENCE

Compact Decomposition of Irregular Tensors for Data Compression: From Sparse to Dense to High-order Tensors

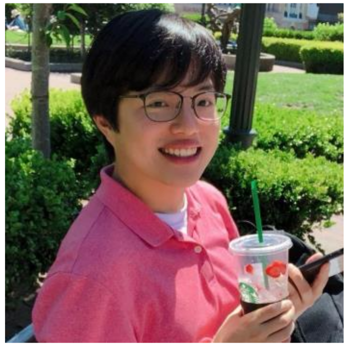
Taehyung Kwon

Jihoon Ko



Jinhong Jung

Jun-Gi Jang

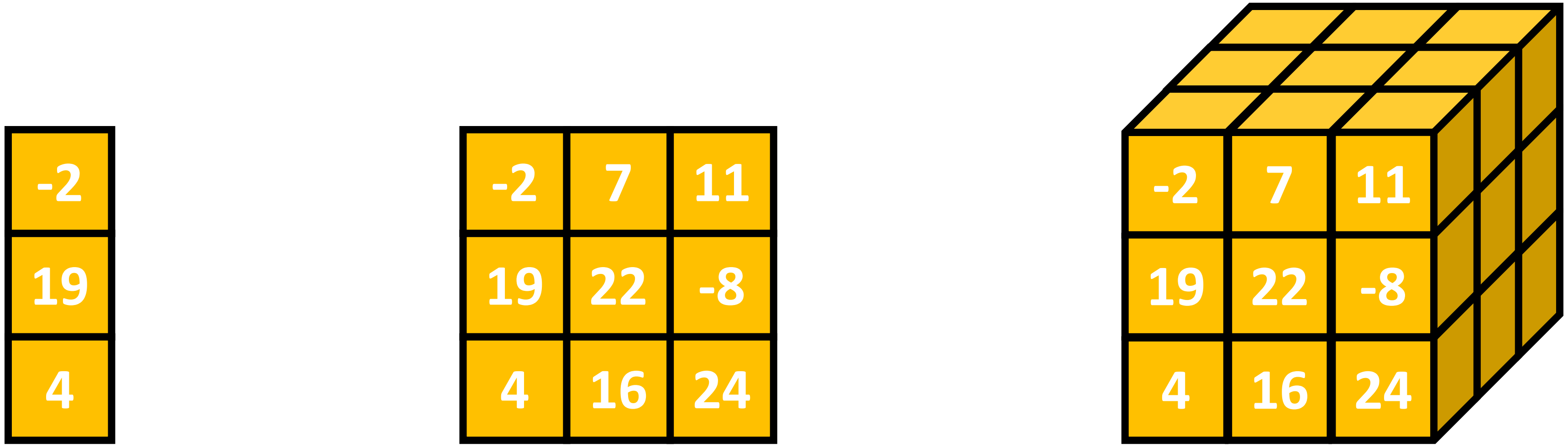


Kijung Shin

Tensor

A **(regular) tensor** is a **multi-dimensional** array.

- In this work, we assume tensors of real values.



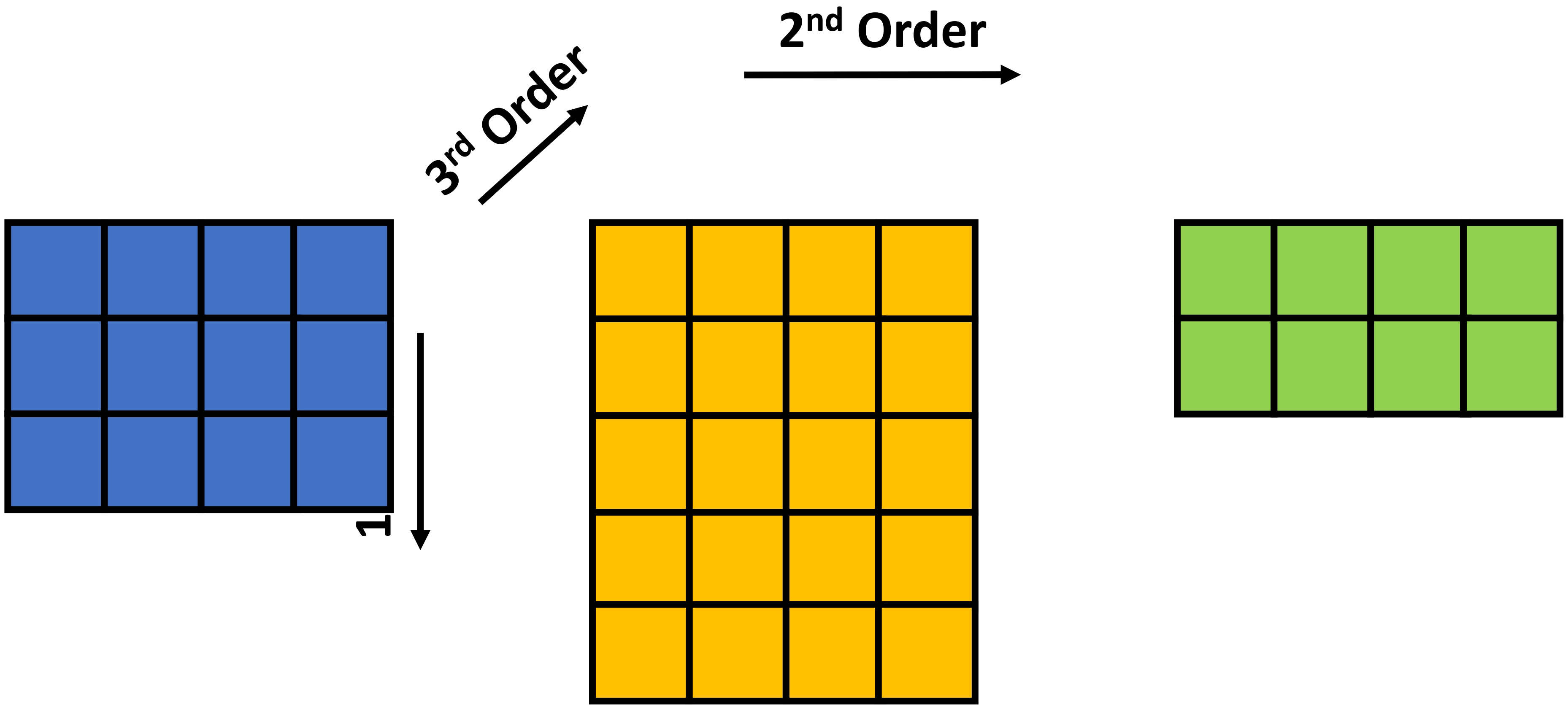
Vector
(1-order tensor)

Matrix
(2-order tensor)

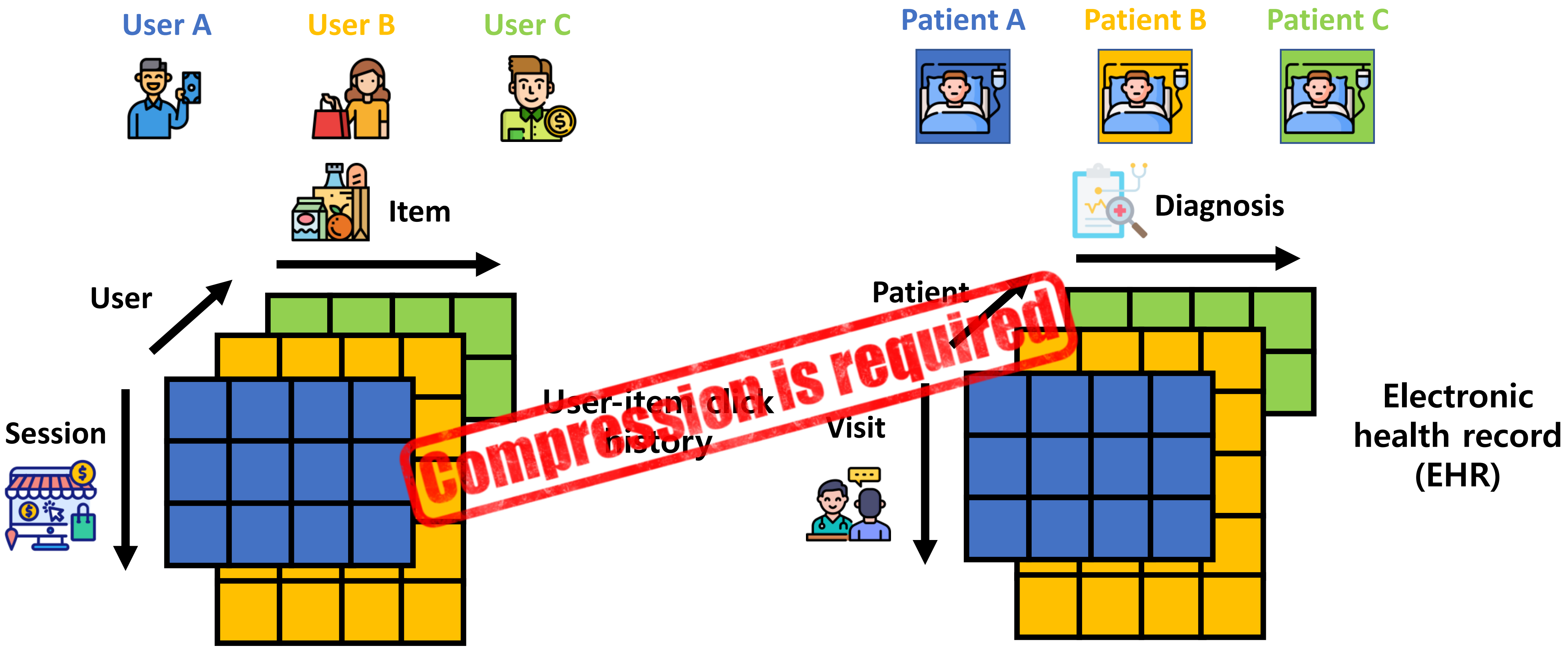
Tensor
(3-order tensor)

Irregular tensor

A (3-order) **irregular tensor** is a collection of matrices with **varying row counts.**



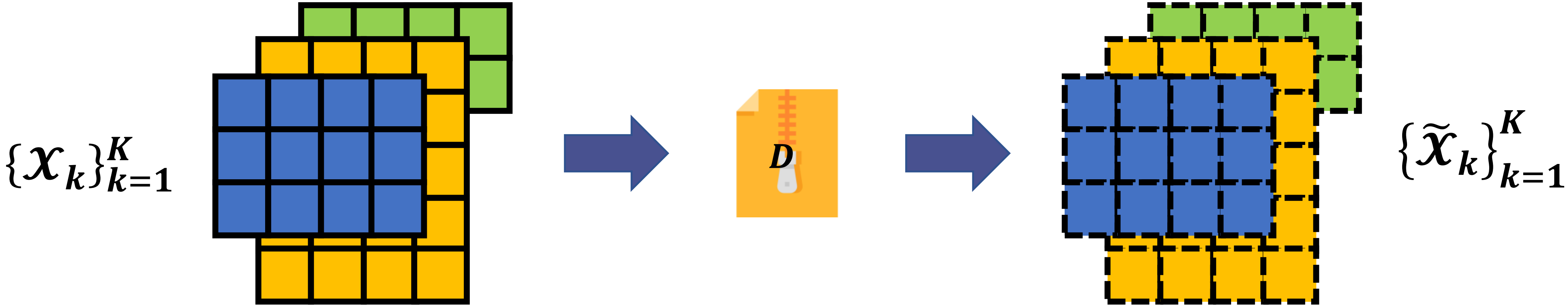
Various real-world data are irregular tensors



More than 140 billion entries

Problem definition

Lossy compression of an irregular tensor.



- **Given**: an irregular tensor $\{\mathcal{X}_k\}_{k=1}^{K}$
- **Find**: the compressed data $\boldsymbol{D}$
- **To minimize**: (1) the size of $\boldsymbol{D}$ and the approximation error $\sum_{k=1}^{K}\left\|\mathcal{X}_k - \widetilde{\mathcal{X}}_k\right\|_F^2$

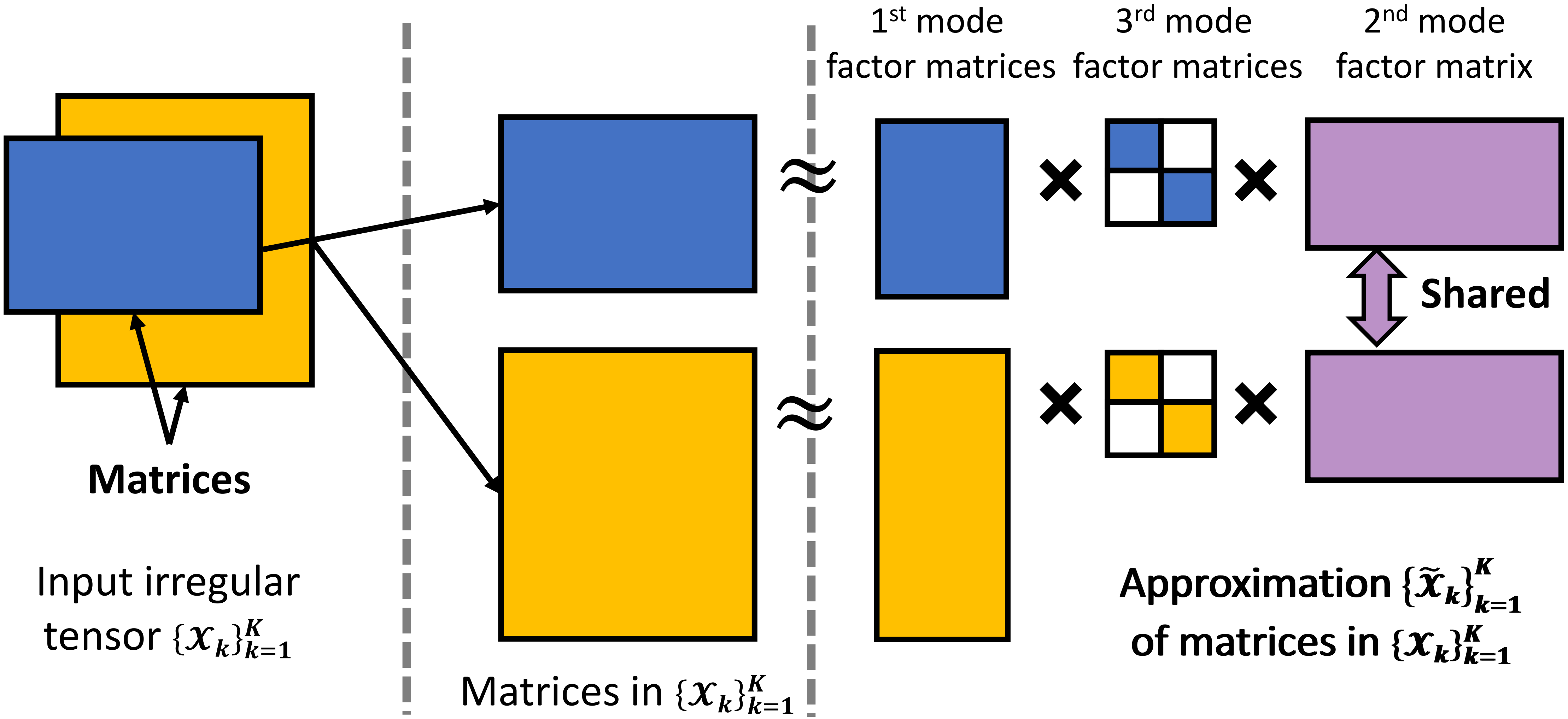
  where $\{\widetilde{\mathcal{X}}_k\}_{k=1}^{K}$ is the approximation of the input tensor.

Outline

1. Introduction.
2. Preliminaries.
3. Proposed method.
4. Experiments.
5. Conclusion.

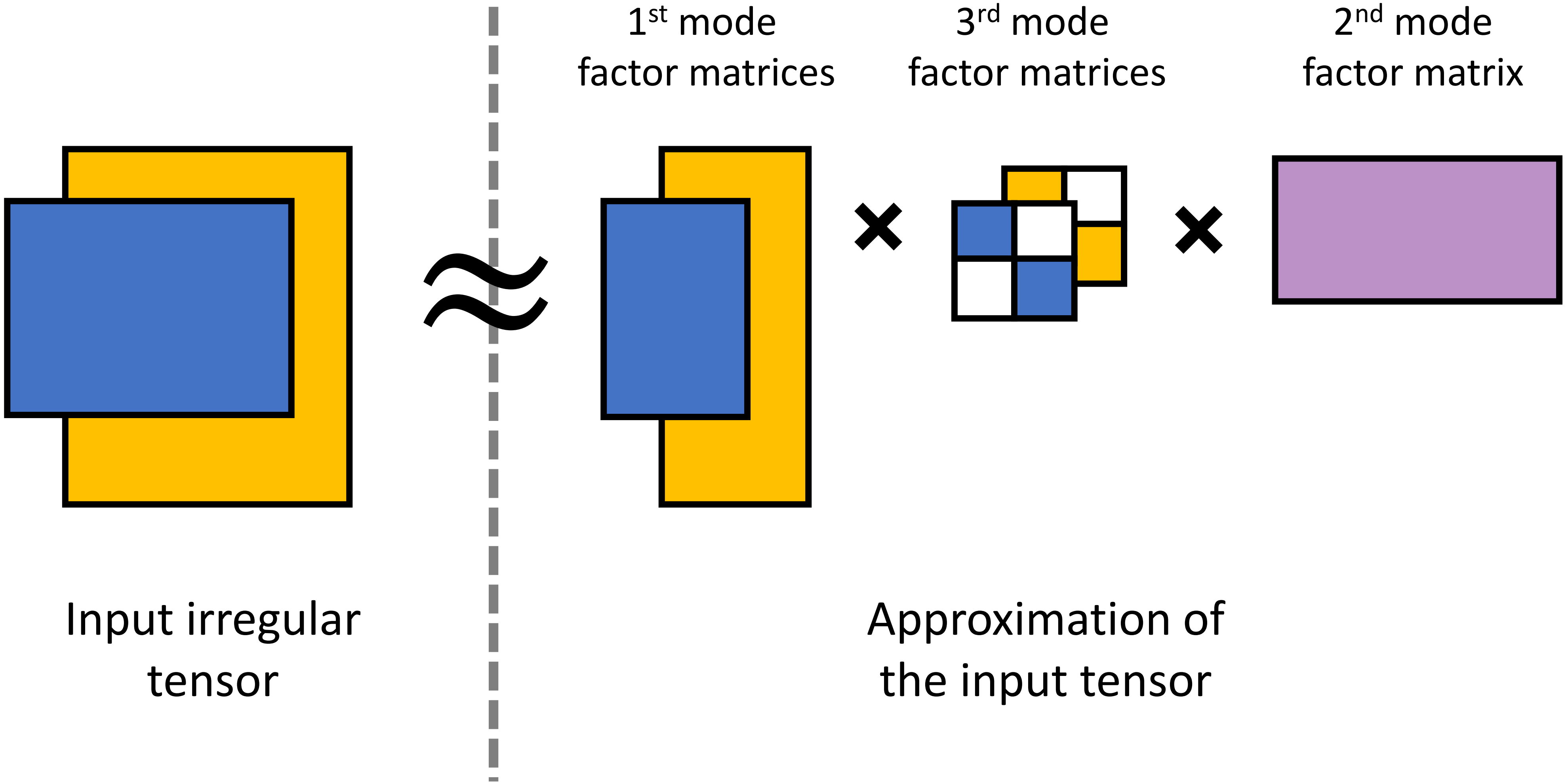
Previous lossy compression: PARAFAC2

PARAFAC2 approximates an irregular tensor with the products of factor matrices.



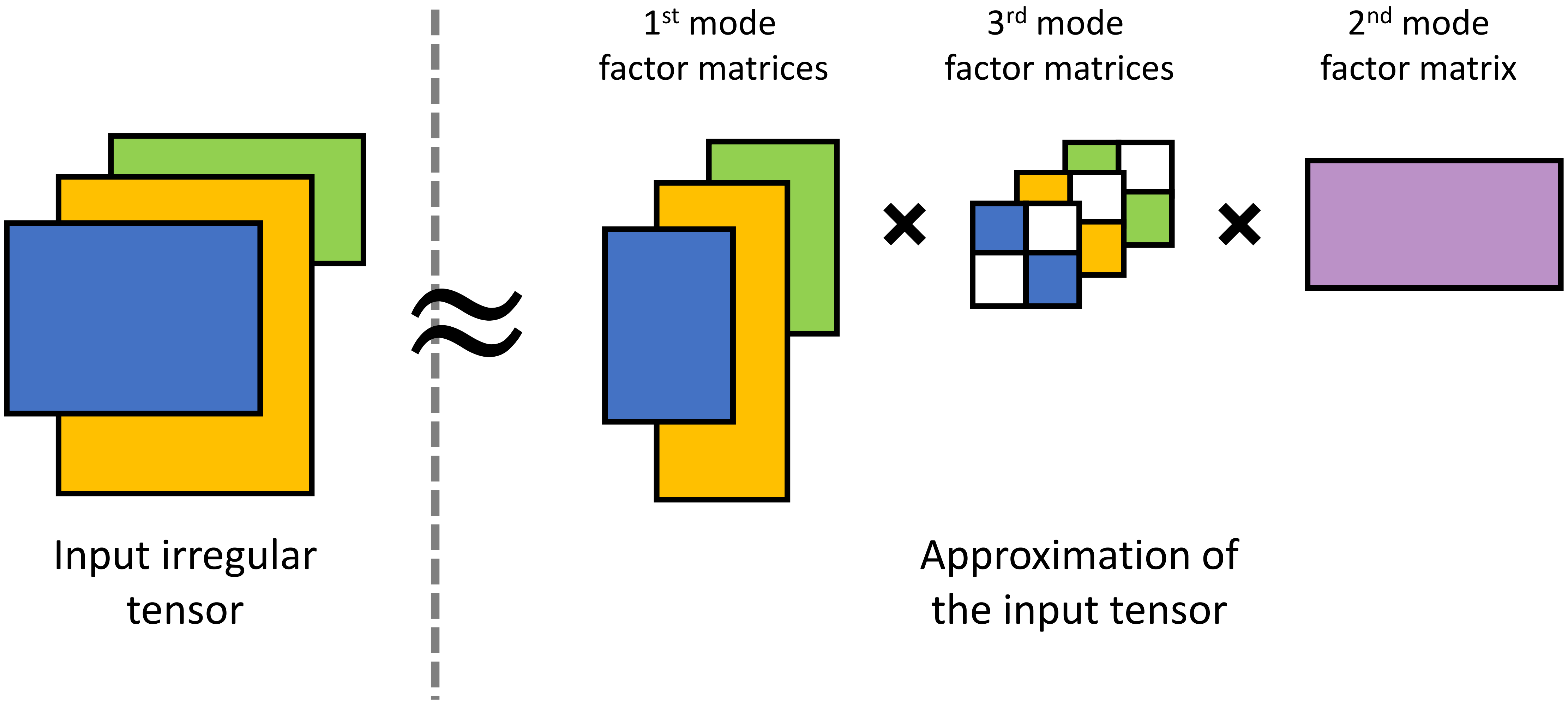
Previous lossy compression: PARAFAC2

PARAFAC2 approximates an irregular tensor with the products of factor matrices.



Previous lossy compression: PARAFAC2

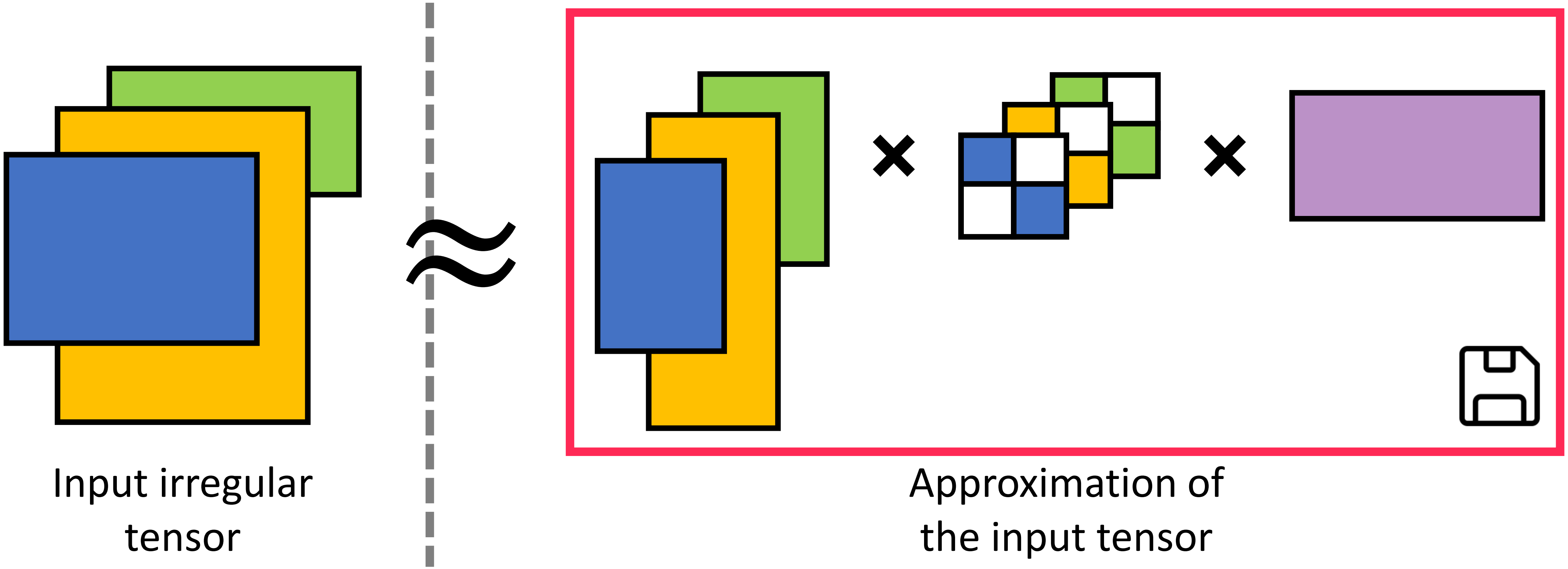
PARAFAC2 approximates an irregular tensor with the products of factor matrices.



Previous lossy compression: PARAFAC2

Factor matrices can be **stored** instead of the input irregular tensor.

That is, factor matrices can be regarded as **the compressed output.**



Input irregular tensor

Approximation of the input tensor

Outline

1. Introduction.
2. Preliminaries.
3. Proposed method.
4. Experiments.
5. Conclusion.

Overview of Light-IT and Light-IT^{++}

We propose **Light-IT and Light-IT^{++}**, lossy compression algorithms for irregular tensors, built upon PARAFAC2.

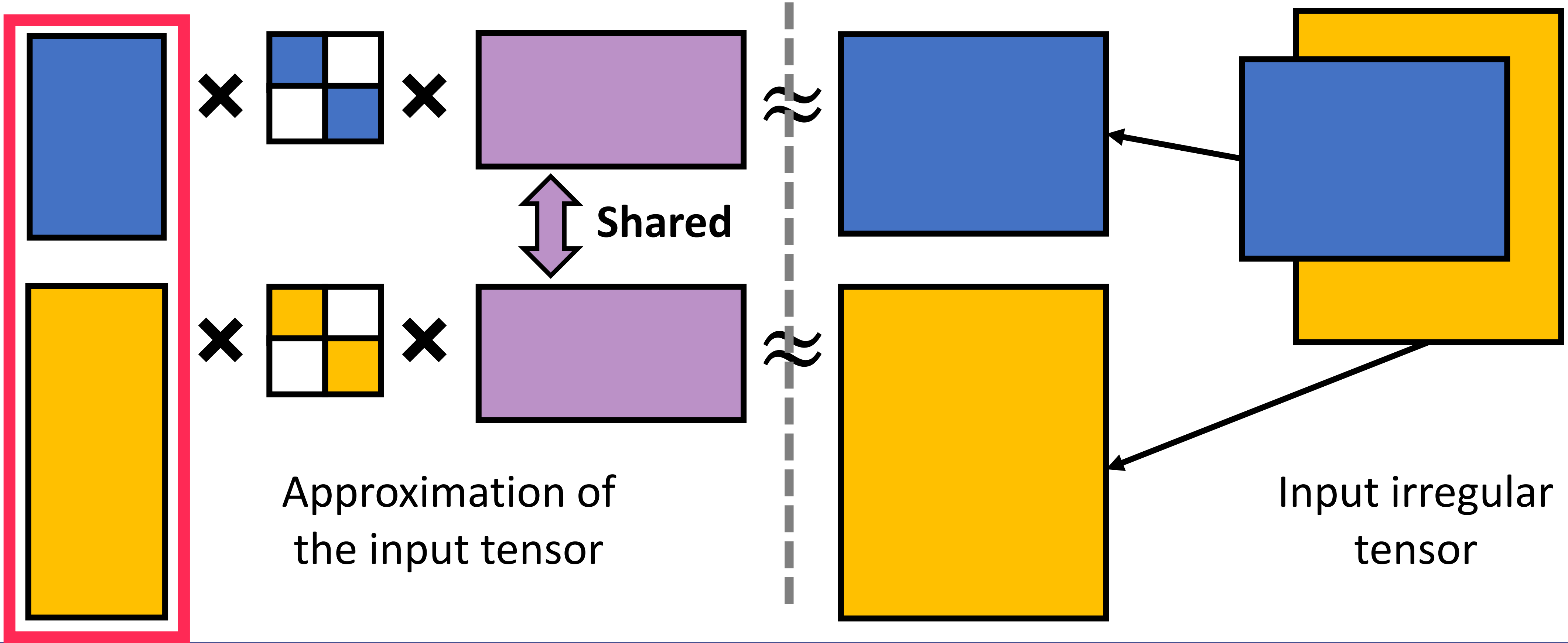
- Q1. Compactness: How can we compress irregular tensors **compactly**?
- Q2. Expressiveness: How can we increase the **expression power** of PARAFAC2?
- Q3. How can we **efficiently** compress **sparse** irregular tensors?
- Q4. How can we compress **higher-order** irregular tensors?

Limited compression ability of PARAFAC2

PARAFAC2 gives a first mode matrix for each slice of an irregular tensor.

- **Bottleneck**: saving all the 1st mode factor matrices is expensive.

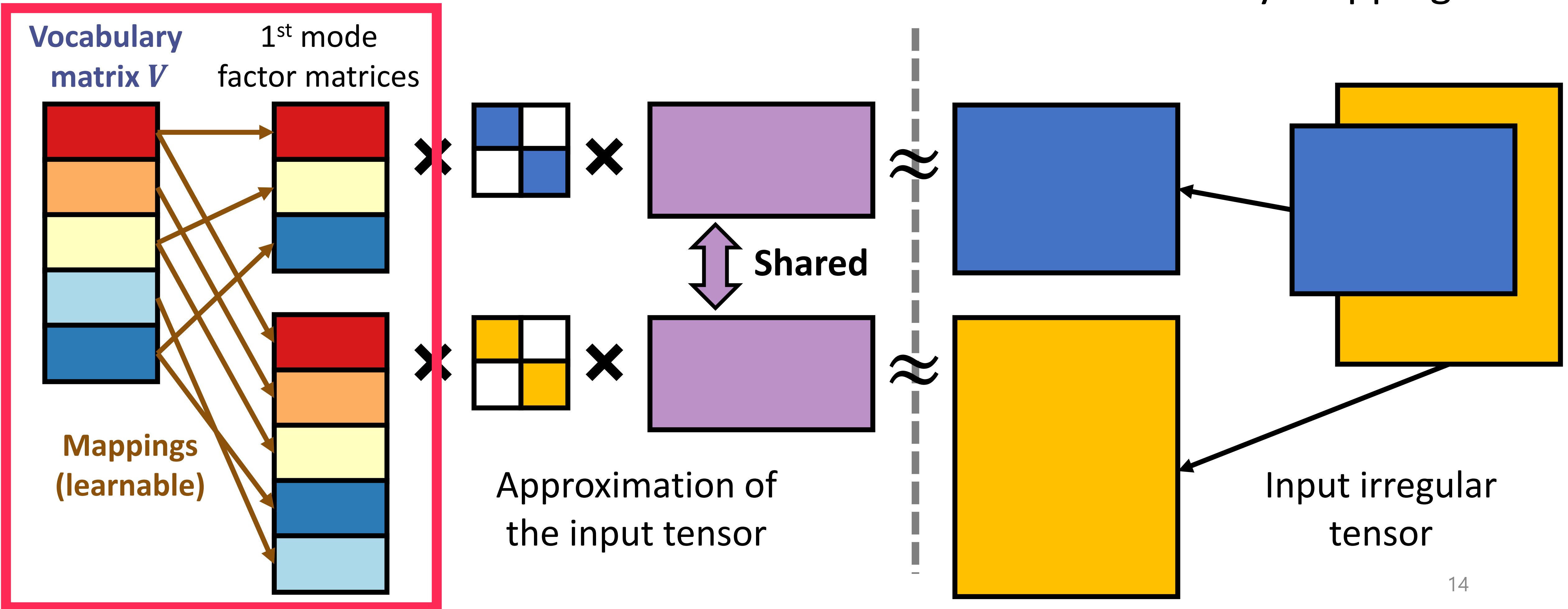
Q1 .How can we make the compression result more **compact**?



A1. Vocabulary-based compression (Light-IT)

We use a single **vocabulary matrix V** shared by all 1st mode factor matrices.

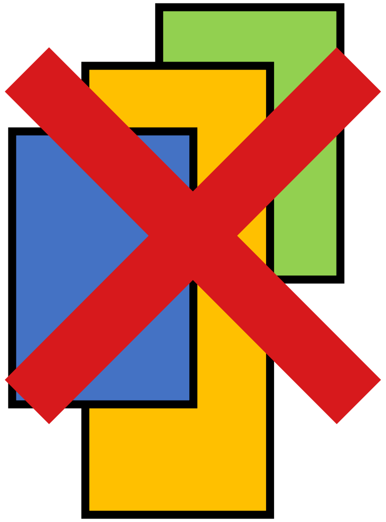
The 1st mode factor matrix for each slice is constructed from V by mappings.



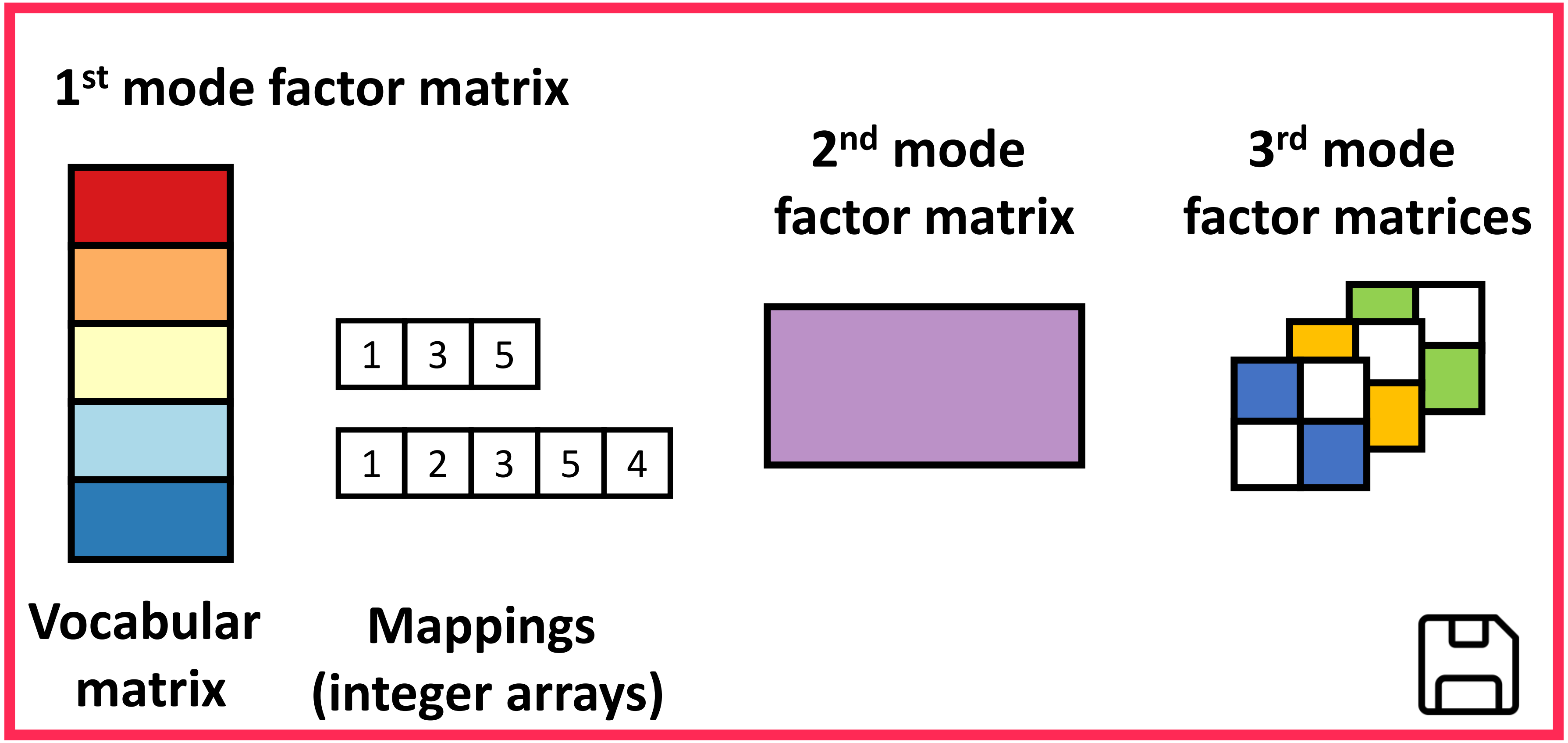
Compression results using Light-IT

Only a single factor matrix per mode is required.

Mappings are further compressed by Huffman encoding.



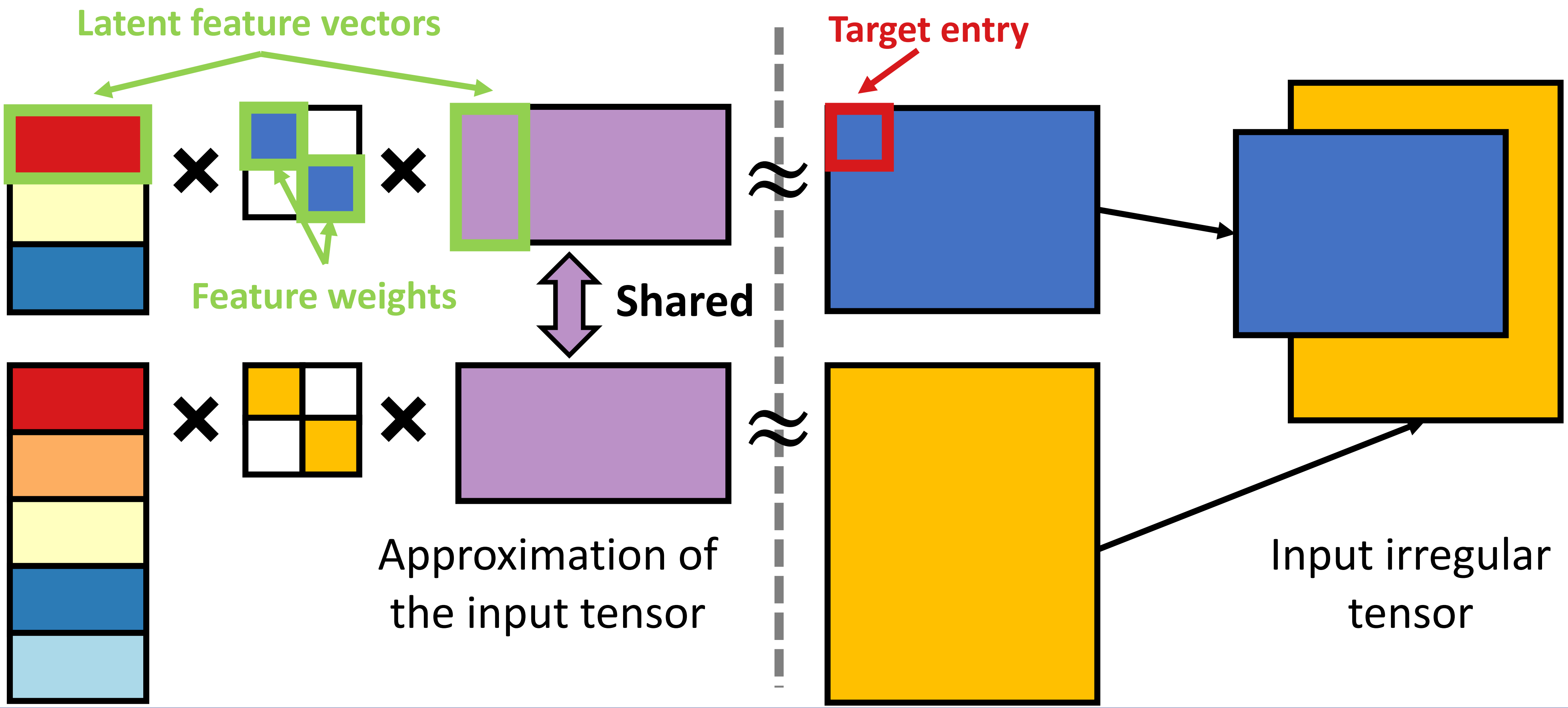
1st mode factor matrices
in PARAFAC2
(not required in Light-IT)



Limited expressional ability of PARAFAC2 (and Light-IT)

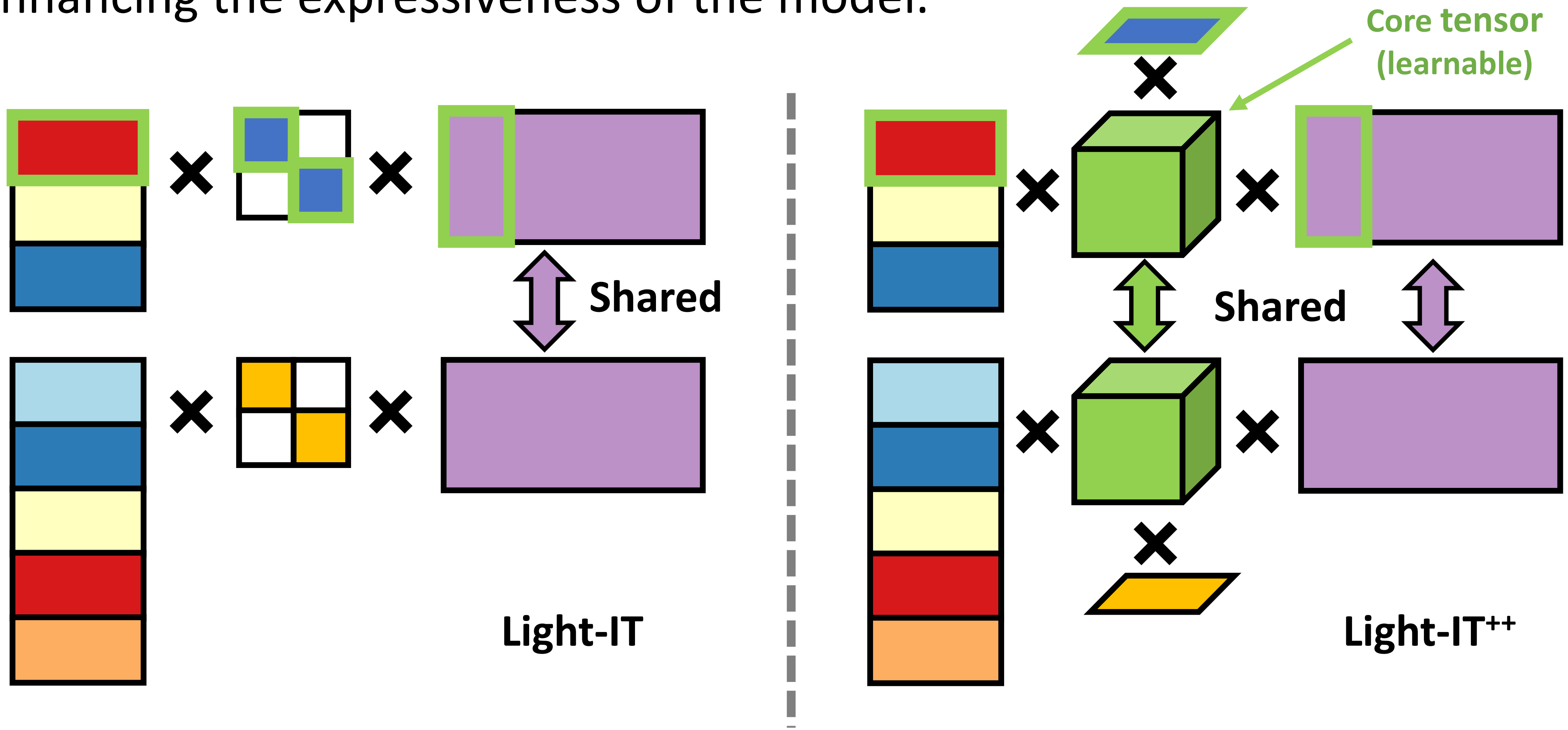
Each **entry** is approximated by the (weighted) product of **feature vectors.**

It fails to capture the relationships between different features.

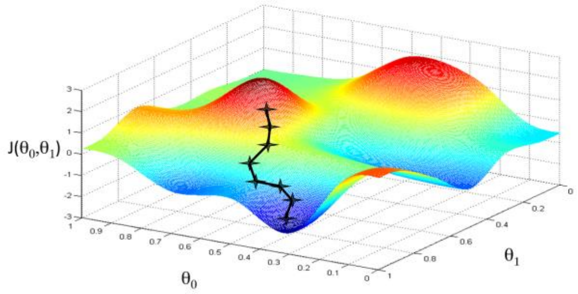


A2. Extension with a core tensor (Light-IT^{++})

Incorporate a **core tensor** to capture relationships between different features, enhancing the expressiveness of the model.



Training of Light-IT and Light-IT^{++}

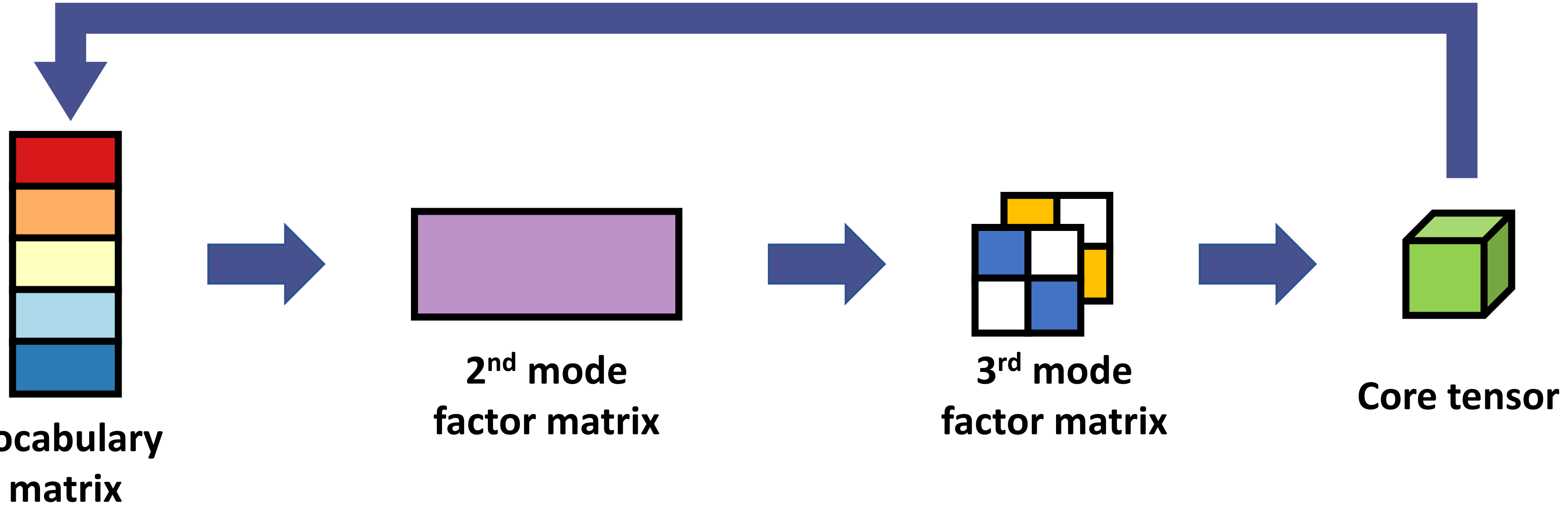


Light-IT: gradient descent to minimize the squared error

- Mappings can be also made differentiable*

Light-IT^{++}: alternating least square (ALS) for sequential updates

- Mappings are fixed to those from Light-IT.



* Differentiable product quantization for end-to-end embedding compression. In ICML 2020.

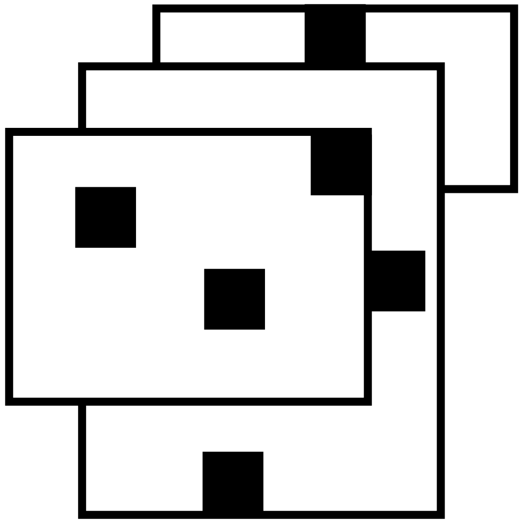
A3. Sparse design

Exploit the **sparsity** of sparse tensors for the efficient computation.

E.g.) complexity of loss computation in Light-IT:

- Naïve computation ∝ **all entry count.**
- Efficient computation ∝ **non-zero entry count**.
- **Key idea:** compute the losses for zero entries efficiently in a closed form

Time complexity ∝

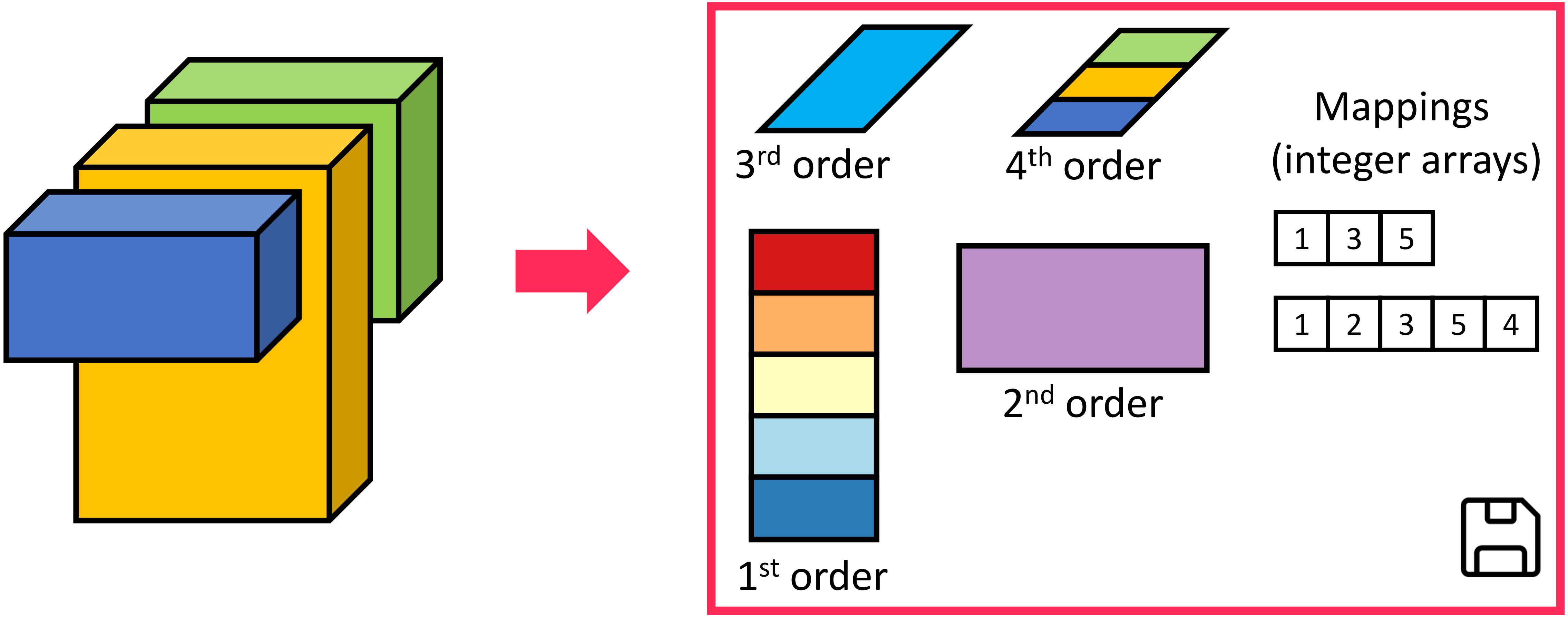


 # Nonzero entries

A4. Higher-order design

Our methods are applicable to irregular tensors of any order.

Key idea: matricize the input irregular tensor and its approximation.



Outline

1. Introduction.
2. Preliminaries.
3. Proposed method.
4. Experiments.
5. Conclusion.

Experimental settings: Datasets

Used 6 public real-world datasets.

- Four 3-order and two 4-order irregular tensors.
- Four sparse and two dense irregular tensors.

Electronic hospital records

Email data

User-interaction data

Stock data

Experimental settings: Baselines

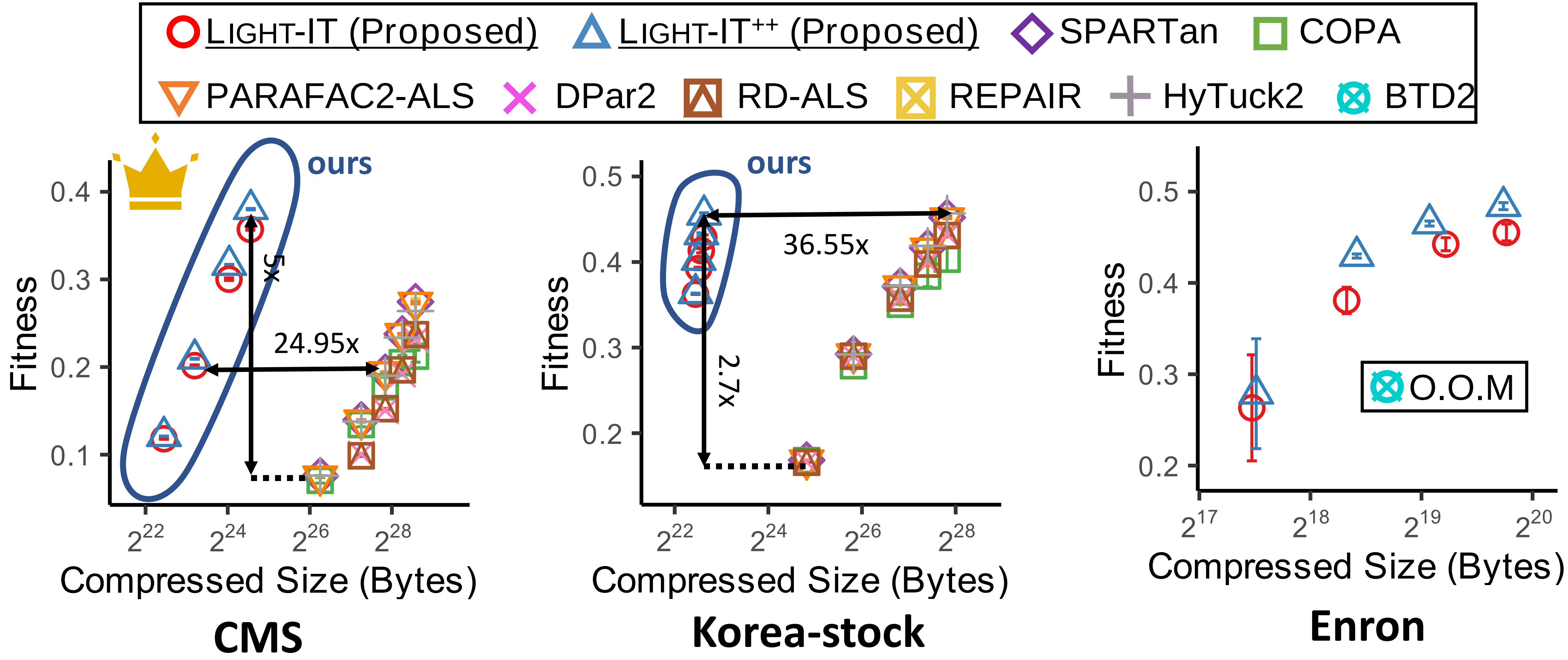
We used lossy-compression baselines.

- Methods for **3-order dense** irregular tensors.
  - PARAFAC2-ALS, RD-ALS, DPar2, and HyTuck2.
- Methods for **3-order sparse** irregular tensors.
  - COPA, SPARTan, and REPAIR.
- A method for **4-order dense** irregular tensors.
  - BTD2.

Our methods are concise and precise

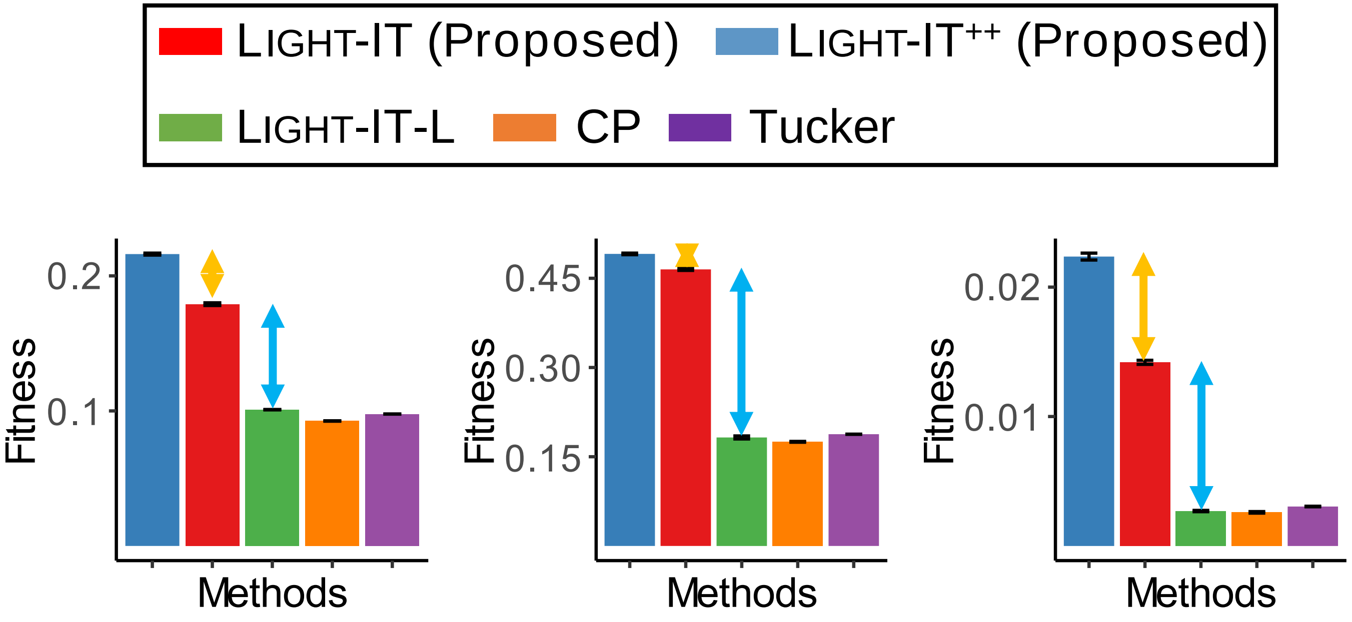
The compressed outputs of our methods are up to 37x smaller.

Our methods show up to 5x better accuracy.



All components of our methods are useful

(1) vocabulary-based compression and (2) extension with a core tensor are effective for compression.



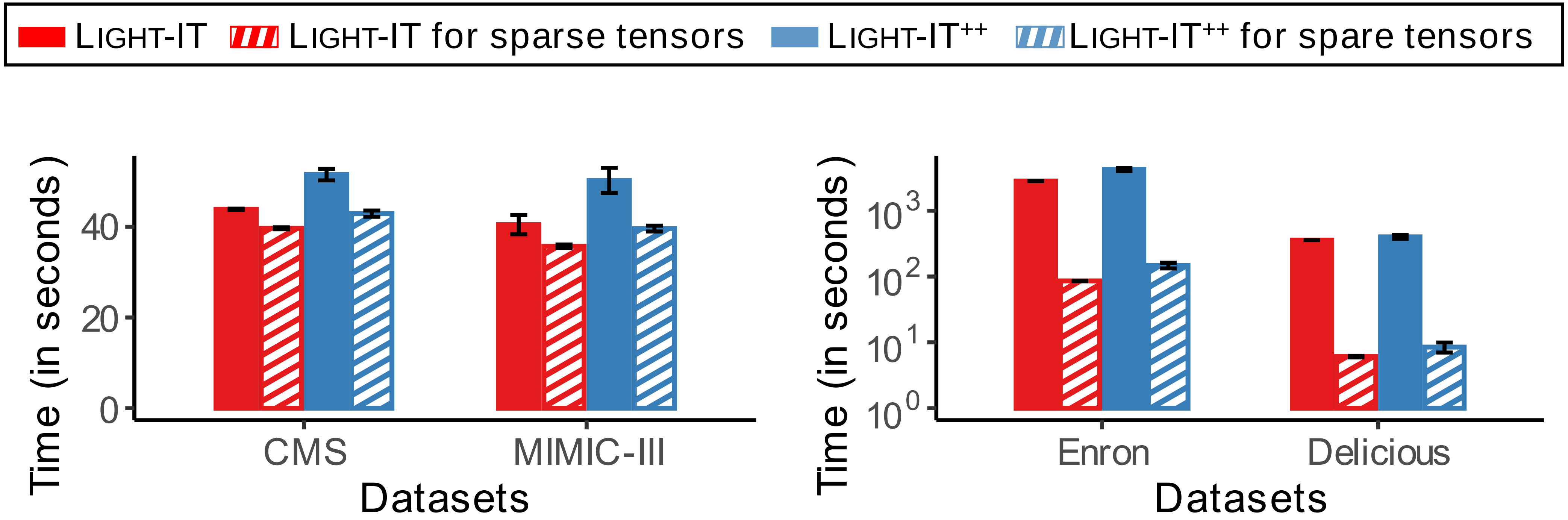
MIMIC-III US-stock Delicious

Datasets

- Light-IT-L: Light-IT variant mapping the i-th row of a slice to the i-th row of the shared matrix.
- CP: CP decomposition.
- Tucker: Tucker decomposition.

Our methods for sparse tensors are faster

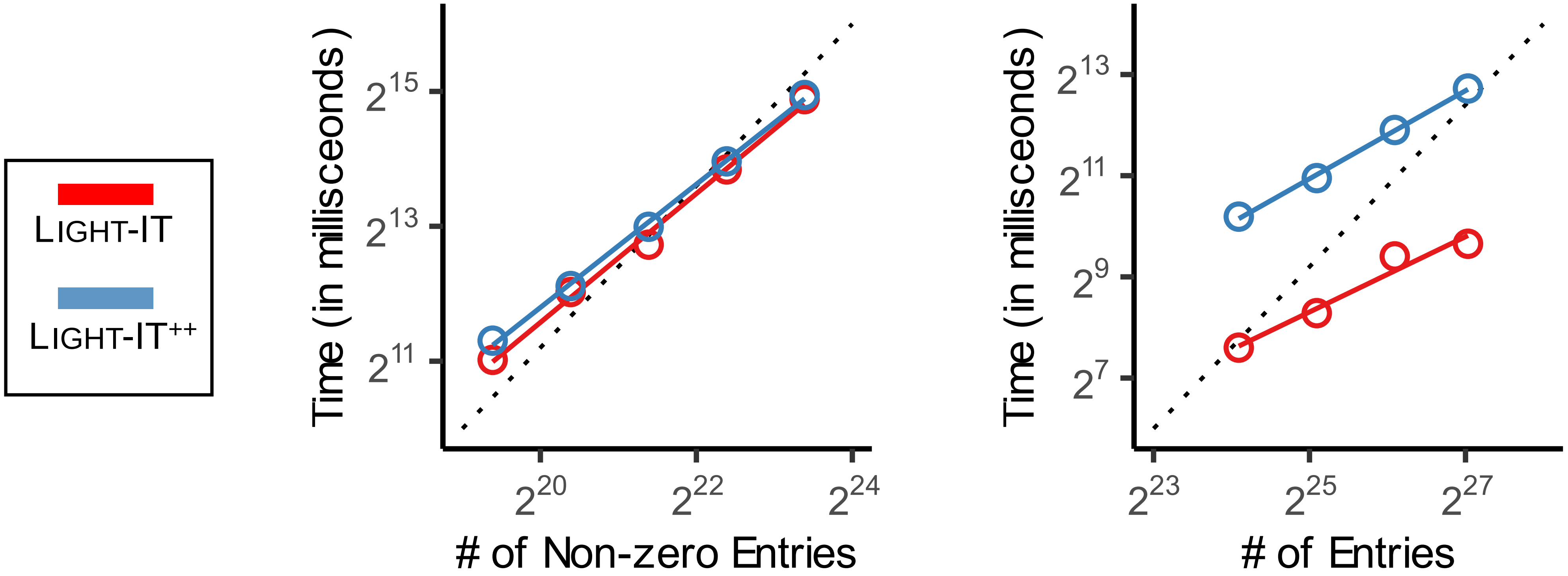
For sparse irregular tensors, the sparse versions of our methods are faster than their dense versions.



3-order irregular tensors **4-order irregular tensors**

Our methods are scalable

Compression time of our methods is linear in the number of (non-zero) entries.



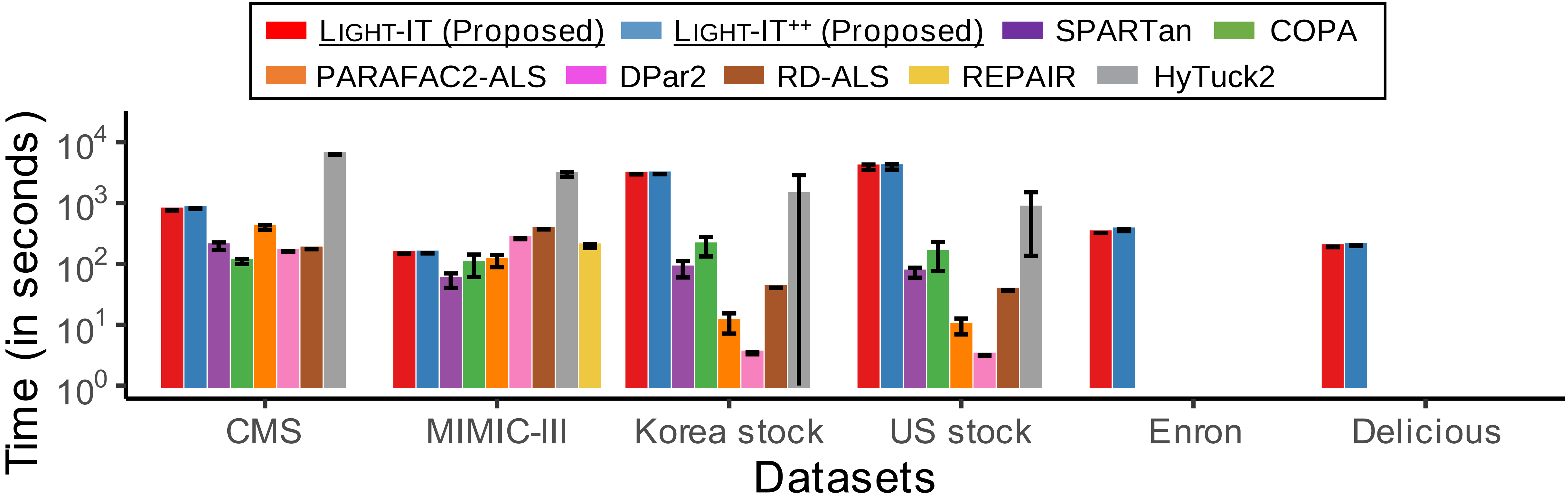
3-order sparse tensors

3-order dense tensors

Total compression time

Our methods tend to be slower than most competitors.

However, our methods took at most 1.1 hours for all considered datasets.



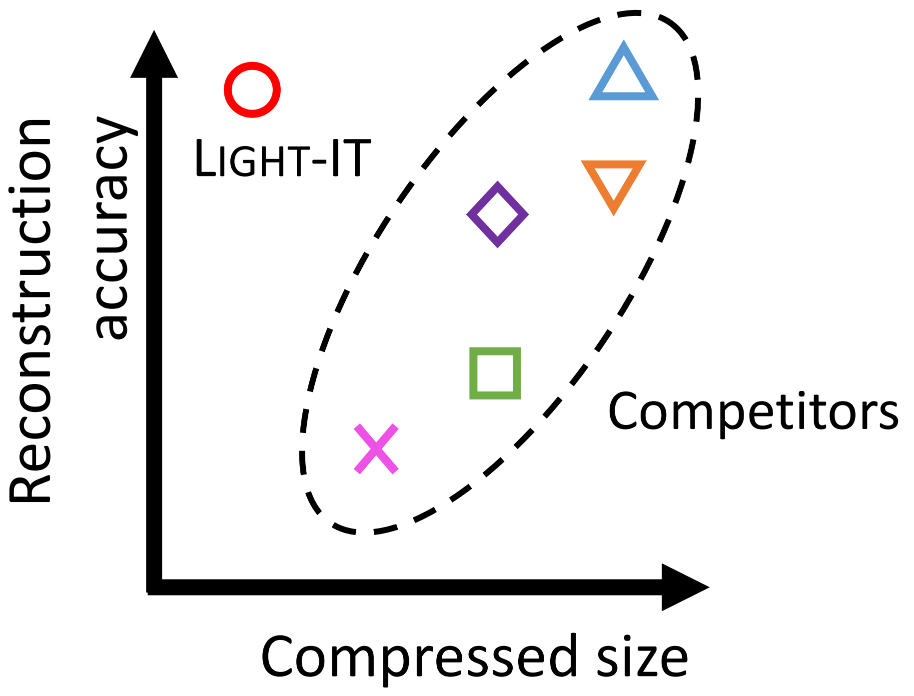
Outline

1. Introduction.
2. Preliminaries.
3. Proposed method.
4. Experiments.
5. Conclusion.

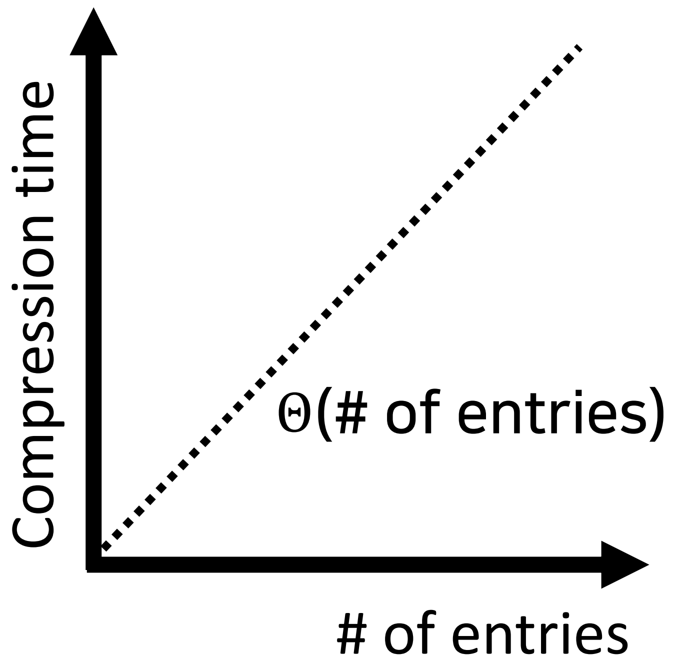
Conclusions

We propose **Light-IT and Light-IT^{++}**, lossy compression algorithms for irregular tensors, built upon PARAFAC2.

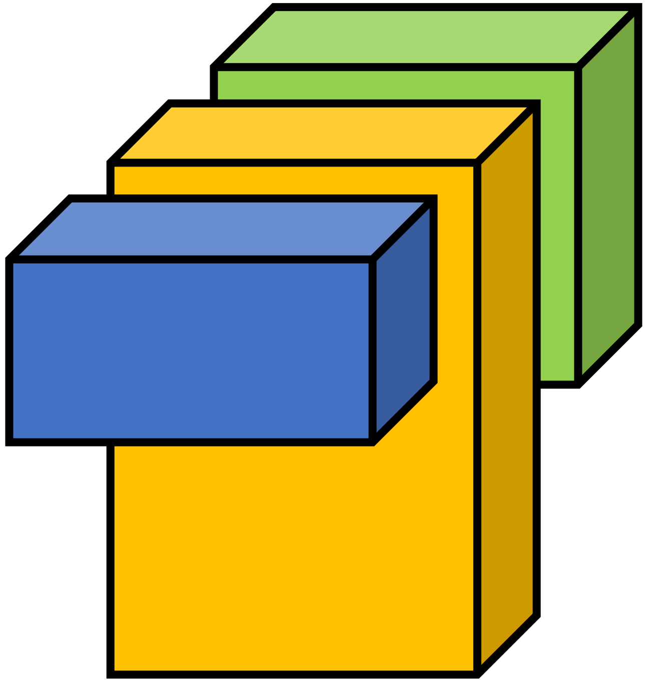
Compact and Accurate



Scalable



Versatile



KDD2024
BARCELONA, SPAIN

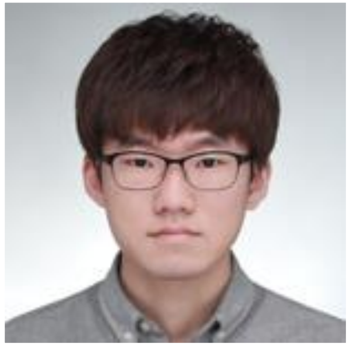
KAIST

COMPUTER SCIENCE

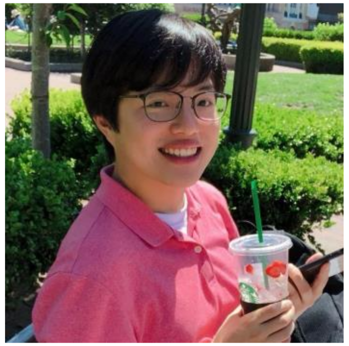
Compact Decomposition of Irregular Tensors for Data Compression: From Sparse to Dense to High-order Tensors


Taehyung Kwon


Jihoon Ko


Jinhong Jung


Jun-Gi Jang


Kijung Shin