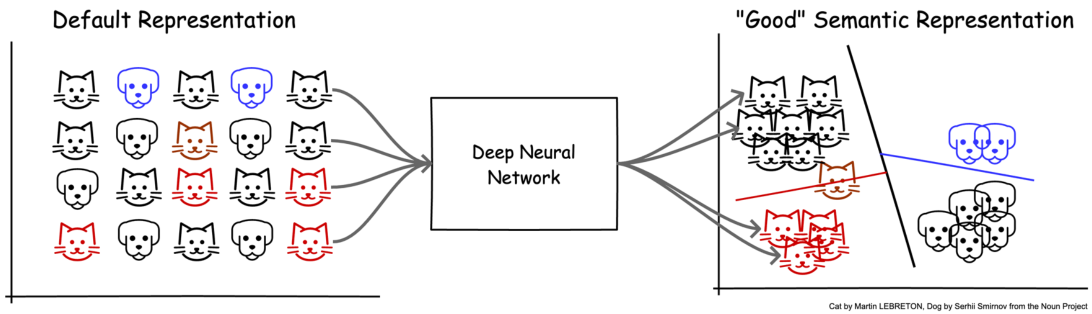
Representation Learning Based on Givens Transformation and its applications

Yanwen Zhang

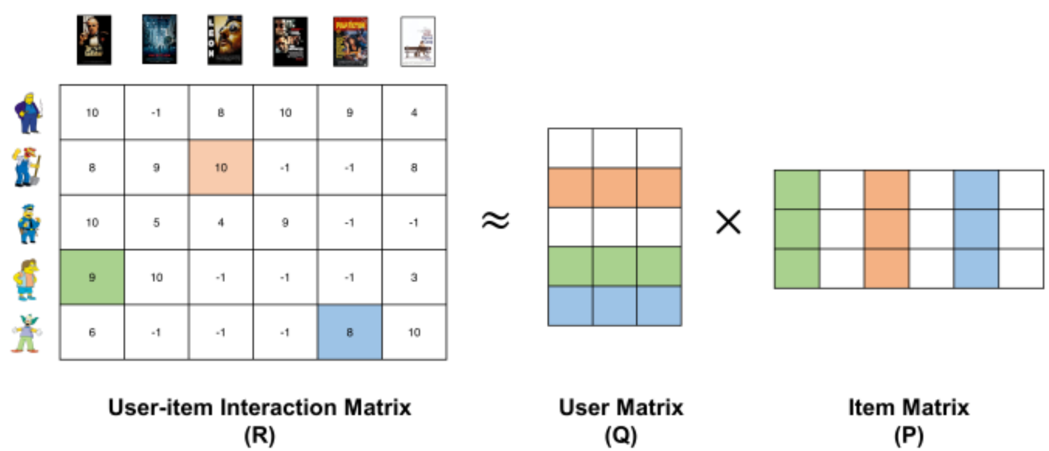
2024.02.23

1 Background

- Representation Learning (RL) involves automatically finding the **features or representations of data** useful for predictive tasks.



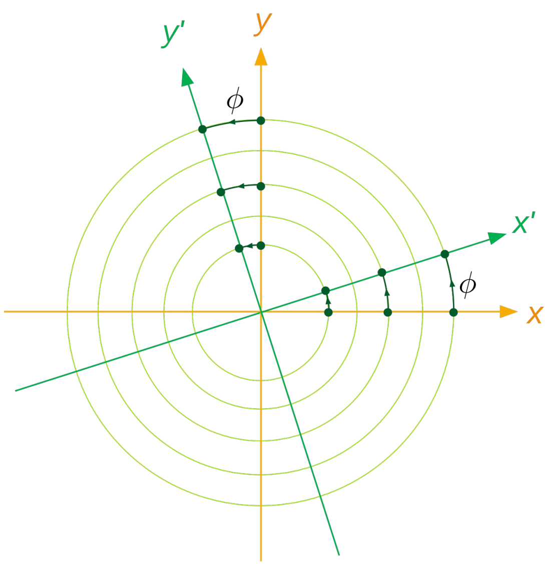
- Matrix Factorization (MF) is a technique to decompose a matrix into a product of matrices, revealing **hidden structures** in data.
- By decomposing data into simpler matrices, it reveals the **underlying structure** or patterns that are essential for representation learning.

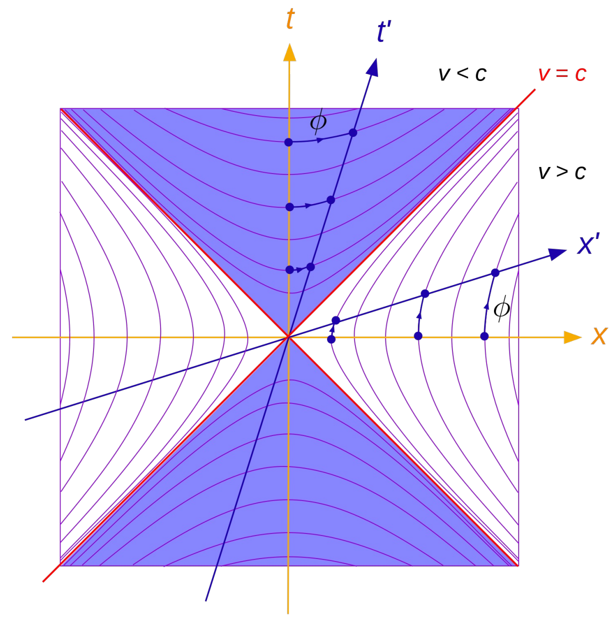


1 Background

- The **“Orthogonal constraints”** of matrix decomposition theories.

- **Advantages**:
  - Ensures the **independence of latent dimensions**;
  - removes **redundant information**;
  - improves the efficiency of representation learning;

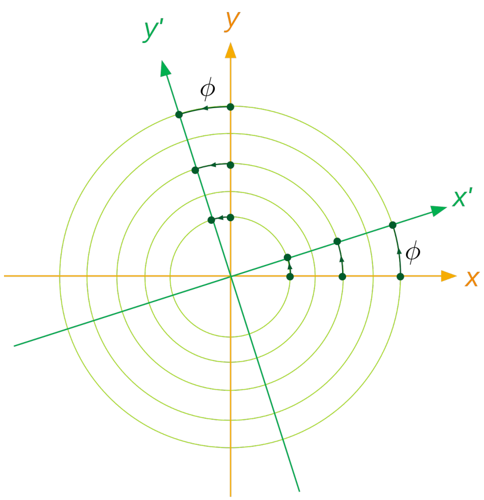


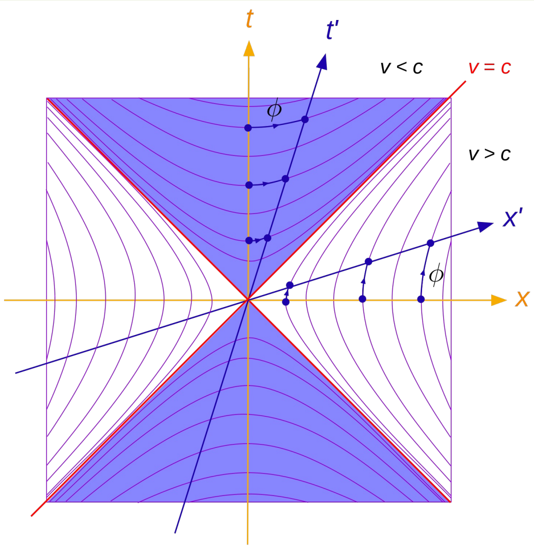


1 Background

- The **“Orthogonal constraints”** of matrix decomposition theories

- **Advantages**:
  - Ensures the **independence of latent dimensions**;
  - removes **redundant information**;
  - improves the efficiency of representation learning;
- **Disadvantages:**
  - When manipulating representations, the orthogonality of the representation space can be easily **disrupted**, affecting the effectiveness of subsequent models;
  - Naturally introduces constraints, thereby **increasing** the number of representational **elements**.





1 Background

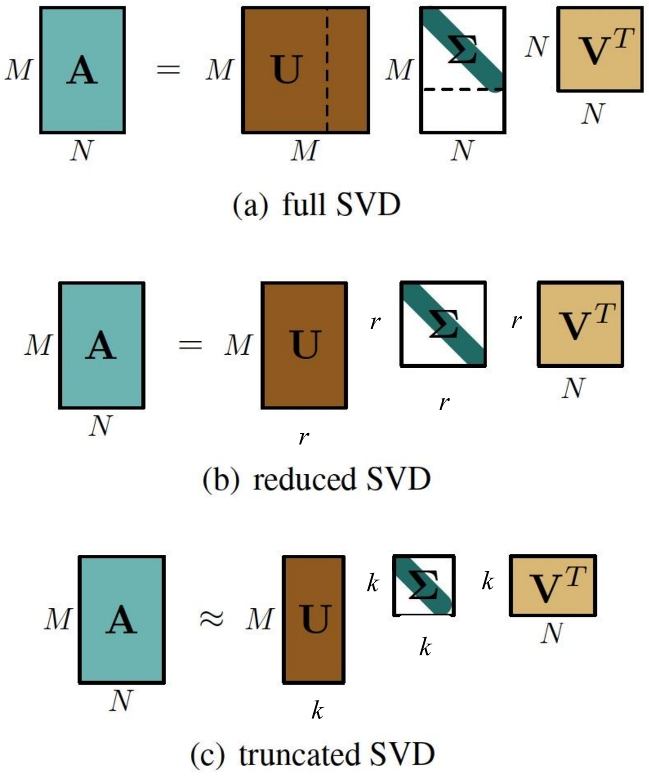
- The “**Orthogonal constraints**” of matrix decomposition theories
- Typical Example: Singular Value Decomposition, SVD

✓ ***U and V*** **have inherent constraints，say** $\boldsymbol{U}$**=**

$[\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_\mathrm{r}]$**:**

$$\boldsymbol{u}_\mathrm{i}\boldsymbol{u}_j = \begin{cases} \mathbf{1}, \boldsymbol{i} = \boldsymbol{j} \\ \mathbf{0}, \boldsymbol{i} \neq \boldsymbol{j} \end{cases}$$

✓ **For column orthogonal matrices with** $\boldsymbol{r}$ **columns, normality introduces** $\boldsymbol{r}$ **constraints, orthogonality introduces** $\boldsymbol{r}(\boldsymbol{r}-\mathbf{1})/\mathbf{2}$ **constraints, thus** $\boldsymbol{r}(\boldsymbol{r}+\mathbf{1})/\mathbf{2}$ **constraints in total.**



(a) full SVD
(b) reduced SVD
(c) truncated SVD

2 Methodology applied

- In order to solve existed problems, we are going to use Givens Transformation to construct the representation space.

- For arbitrary vector $W = (w_1, w_2, \dots, w_m)^T$, $1 \le k < i \le m$, Givens matrix is defined as

$$G_{ki} = \begin{bmatrix} I_{k-1} & 0 & \dots & 0 & 0 \\ 0 & c & \dots & d & 0 \\ \dots & \dots & I_{i-k-1} & \dots & \dots \\ 0 & -d & \dots & c & 0 \\ 0 & 0 & \dots & 0 & I_{m-i} \end{bmatrix}$$

where, $c = \frac{w_k}{s_{ki}} = cos\theta_{ki}, d = \frac{w_i}{s_{ki}} = sin\theta_{ki}$, $(s_{ki} = (w_k^2 + w_i^2)^{1/2})$ posed in (k, i) and (i, k).

- $G_{ki}W$ represents a counterclockwise rotation of the vector x in the (i, j) plane of θ radians：

$$G_{ki}W = (w_1, \dots, w_{k-1}, s_{ki}, \dots, w_{i-1}, 0, \dots)^T;$$

- Thus $G_{1m}G_{1(m-1)} \dots G_{12}W = (s, 0, \dots, 0)^T$, where $s = (w_1^2 + w_2^2 + \dots + w_m^2)^{1/2}$.

- For normalized vector and $k = 1$：

$$G_{1m}G_{1(m-1)} \dots G_{12}W = (1, 0, \dots, 0)^T.$$

2 Methodology applied

- **Givens Transformation of Column Orthogonal Matrices**
- **For column orthogonal matrix $A \in \mathbb{R}^{m\times r}$, there are$[(m\times r) - r(r+1)/2]$Givens matrices(k = 1,2,…,r; i = k+1, k+2,…,m) that satisfy：**

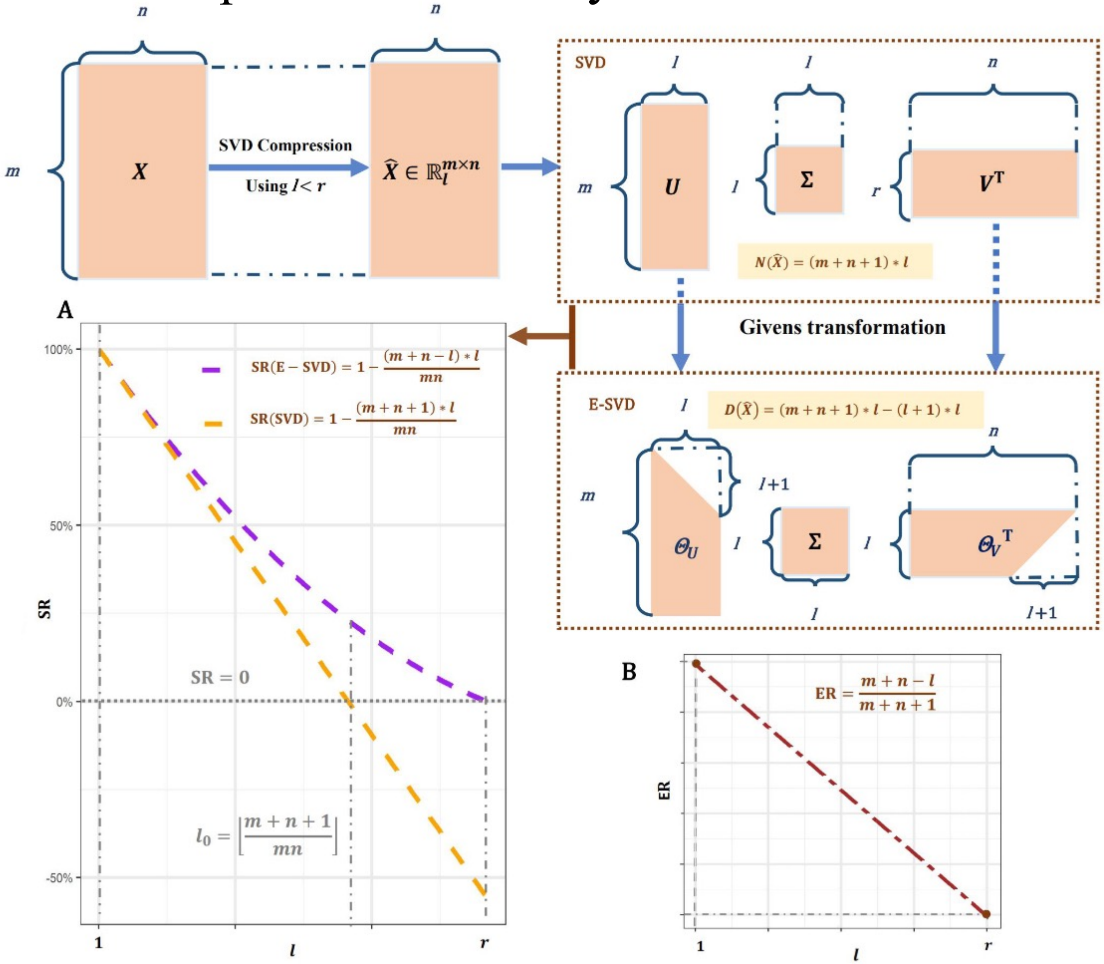
$$(G_{rm}G_{r(m-1)}\ldots G_{r(r+1)})\ldots(G_{2m}G_{2(m-1)}\ldots G_{23})(G_{1m}G_{1(m-1)}\ldots G_{12})A = \begin{bmatrix} I_r \\ O_{(m-r)\times r} \end{bmatrix},$$

- **Inverse Transformation of Column Orthogonal Matrices**
- **Storing $[(m\times r) - r(r+1)/2]$Givens matrices，we can get $A \in \mathbb{R}^{m\times r}$ ：**

$$A = (G_{12}^{\mathrm{T}}G_{13}^{\mathrm{T}}\ldots G_{1m}^{\mathrm{T}})\ldots(G_{r(r+1)}^{\mathrm{T}}G_{r(r+2)}^{\mathrm{T}}\ldots G_{rm}^{\mathrm{T}})\begin{bmatrix} I_r \\ O_{(m-r)\times r} \end{bmatrix}.$$

3 Constructing embedding space

- Based on that, we can easily acquire an enhanced version of SVD (**E-SVD**), a method eliminating all redundancies in SVD matrices and enhancing SVD compression losslessly.



- (1) The number of storage units SVD compression:

$$(m+n+1)\cdot l;$$

- (2) The freely valued elements of orthonormal column matrix $\boldsymbol{U}$:

$$m\cdot l-0.5^{*}\, l\cdot(l+1);$$

- (3) The freely valued elements of orthonormal column matrix $\boldsymbol{V}$:

$$n\cdot l-0.5^{*}\, l\cdot(l+1);$$

- (4) The nonzero elements of diagonal matrix $\boldsymbol{\Sigma}$: l;

- (5) The number of storage units after E-SVD:

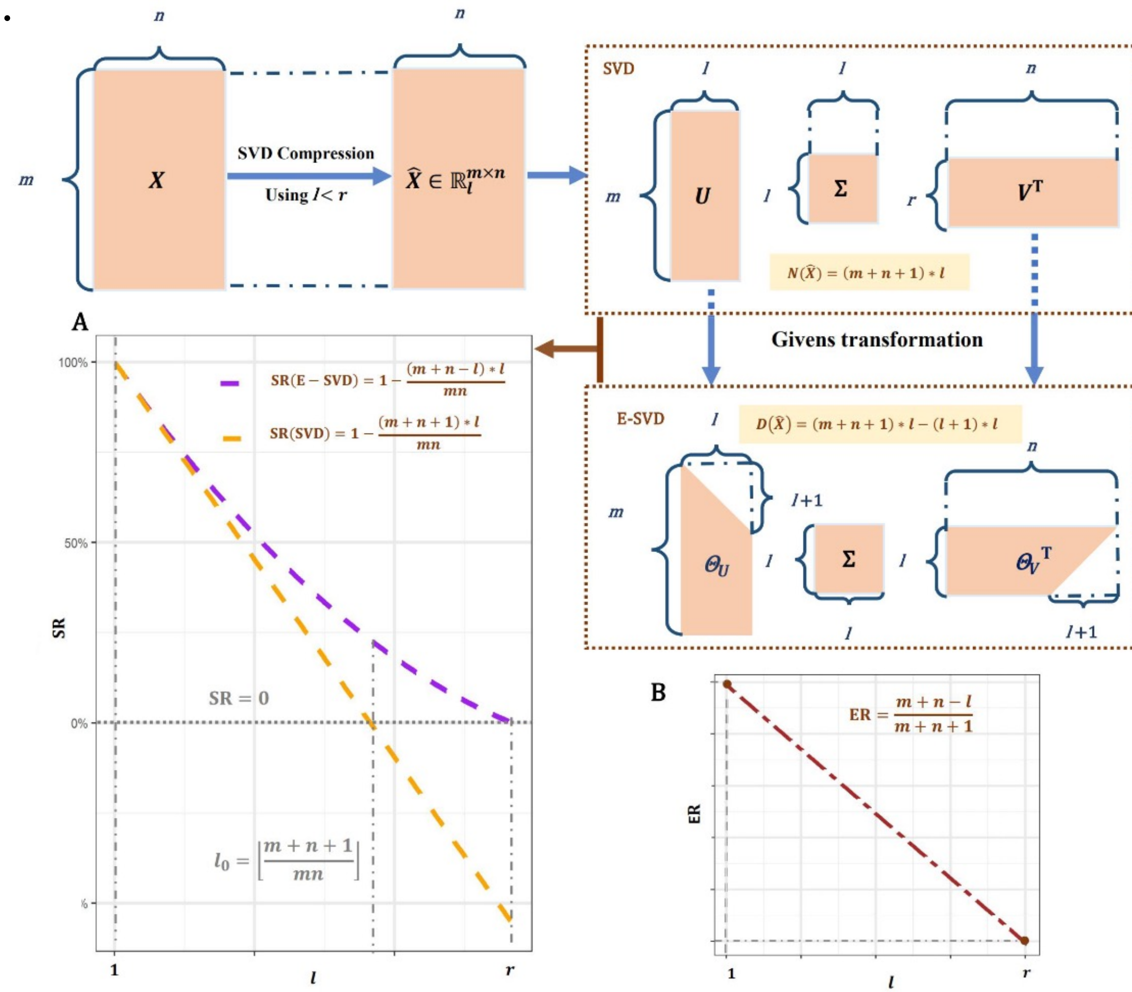
$$(m+n-l)\cdot l.$$

[1] Wang H, Zhang Y, Zhao J. Enhancing the SVD compression losslessly[J]. Journal of Computational Science, 2023, 74: 102182.

3 Constructing embedding space

Storage ratio analysis

- We can use Storage Ratio (SR) to indicate the ratio of the reduced storage units to that of the original matrix.



[1] Wang H, Zhang Y, Zhao J. Enhancing the SVD compression losslessly[J]. Journal of Computational Science, 2023, 74: 102182.

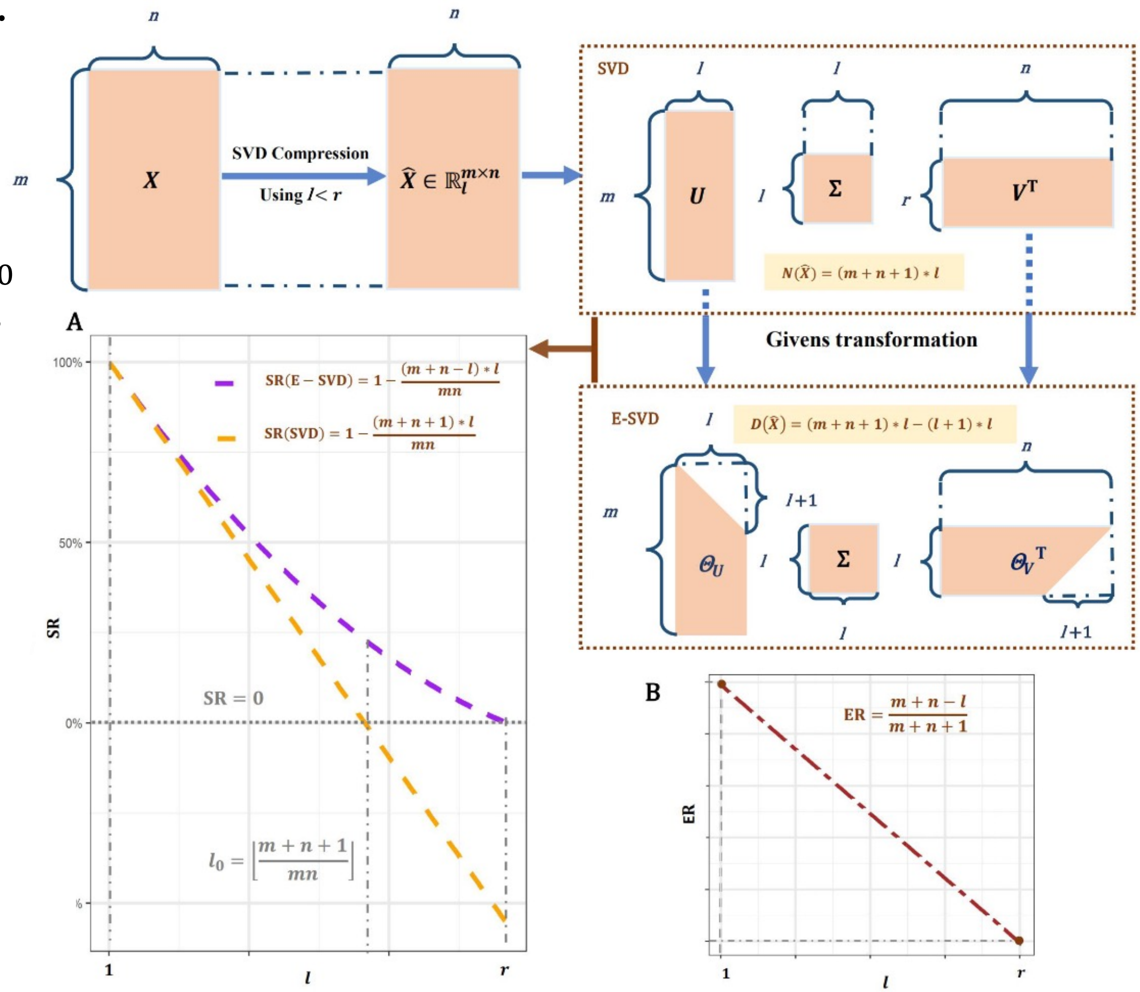
3 Constructing embedding space

Storage ratio analysis

- We can use Storage Ratio (SR) to indicate the ratio of the reduced storage units to that of the original matrix.

✓ The SVD compression would fail to compress data in some situations.

✓ To clearer refer to the limitation of SVD compression, we would denote l_0 as the l where SVD compression fails, which satisfies $l_0 = \lfloor \frac{m \cdot n}{m+n+1} \rfloor$.



[1] Wang H, Zhang Y, Zhao J. Enhancing the SVD compression losslessly[J]. Journal of Computational Science, 2023, 74: 102182.

3 Constructing embedding space

Storage ratio analysis

- We can use Storage Ratio (SR) to indicate the ratio of the reduced storage units to that of the original matrix.

✓ The SVD compression would fail to compress data in some situations.

✓ To clearer refer to the limitation of SVD compression, we would denote $l0$ as the l where SVD compression fails, which satisfies $l_0 = \lfloor \frac{m \cdot n}{m+n+1} \rfloor$.

✓ **When SVD fails, E-SVD can still compress the data and use only 75% storage space to preserve the same amount of information.**

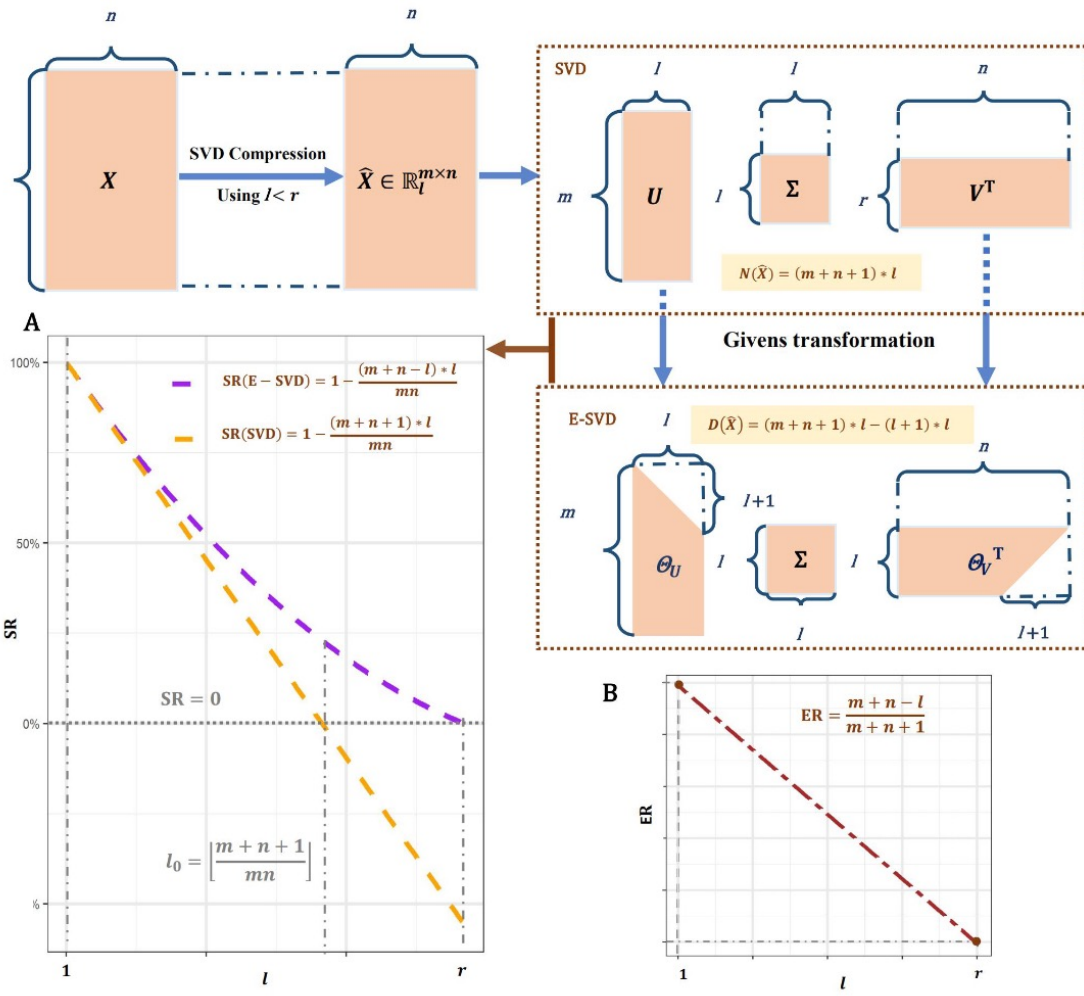
Proof. For $X \in \mathbb{R}_r^{m \times n}$, when $m = n$ and $l = \frac{m \cdot n}{m+n+1}$,

$$\lim_{m \to +\infty} \text{ER} = \lim_{m \to +\infty} [\frac{2m - \frac{m^2}{2m+1}}{2m+1}]$$

$$= \lim_{m \to +\infty} [\frac{2m}{2m+1} - \frac{m^2}{(2m+1)^2}].$$

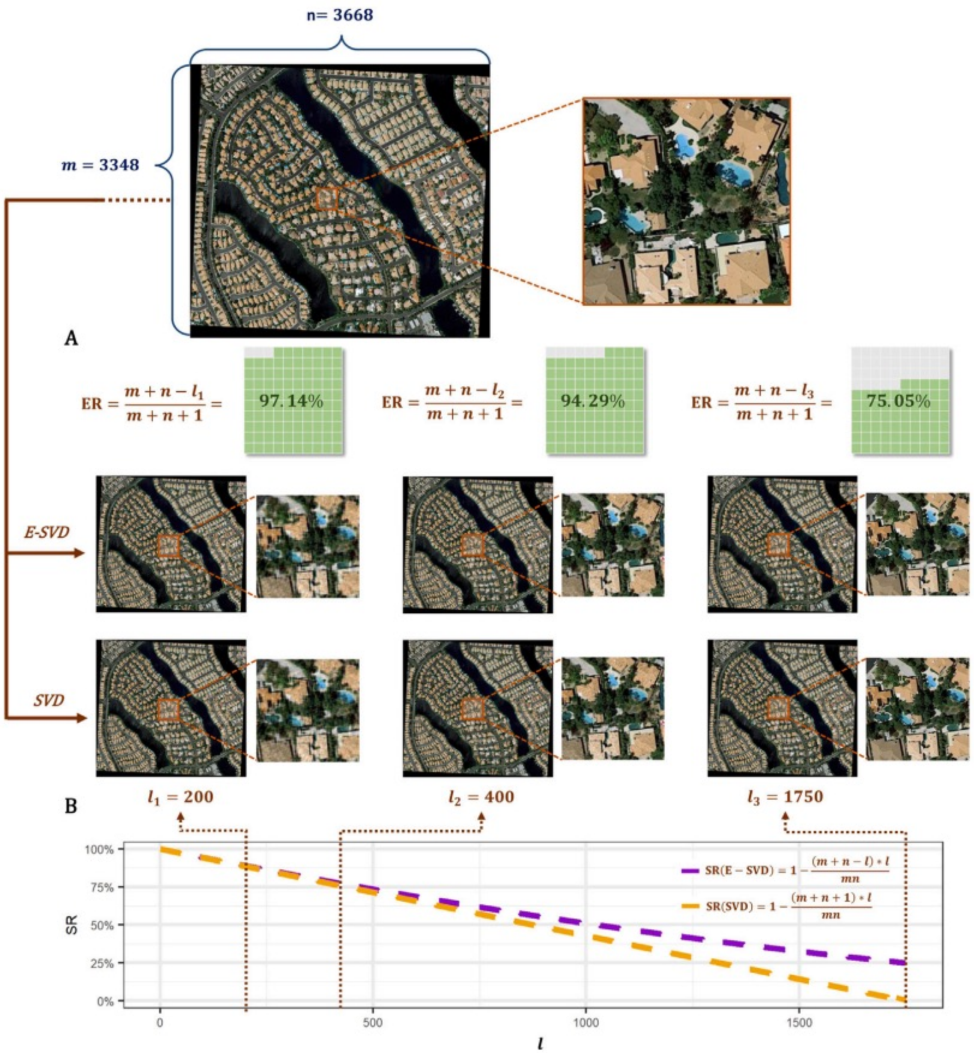
Using L'Hospital's rule, we can get

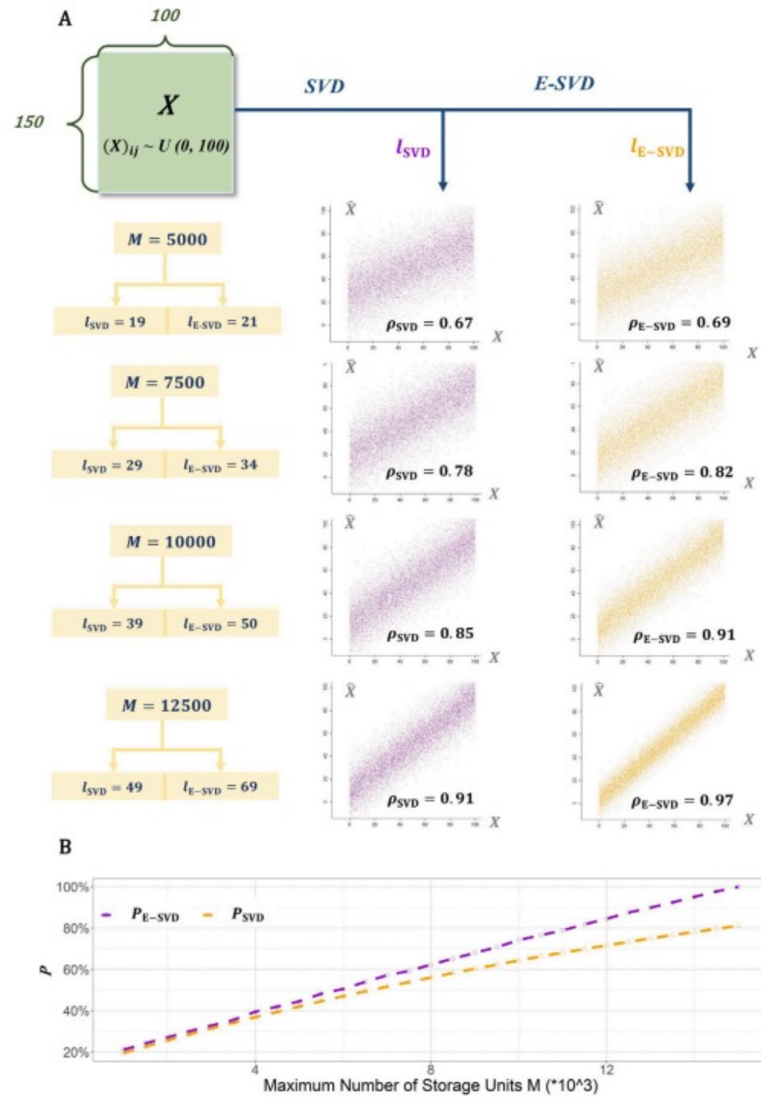
$$\lim_{m \to +\infty} \text{ER} = 1 - \frac{2m}{4(2m+1)} = 1 - \frac{1}{4} = \frac{3}{4} = 75\%.$$



3 Constructing embedding space

Experimental Evidences

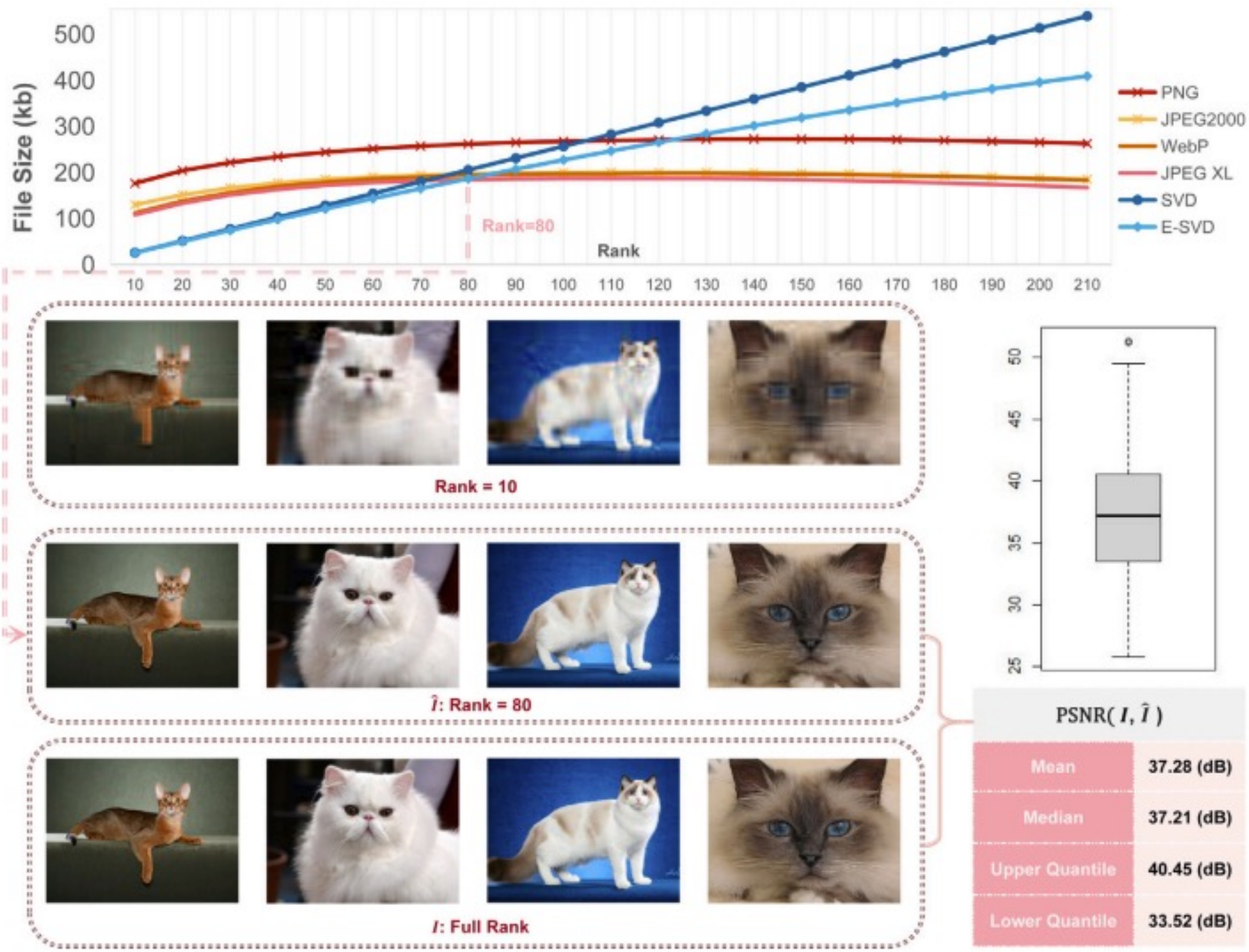




[1] Wang H, Zhang Y, Zhao J. Enhancing the SVD compression losslessly[J]. Journal of Computational Science, 2023, 74: 102182.

3 Constructing embedding space

Experimental Evidences

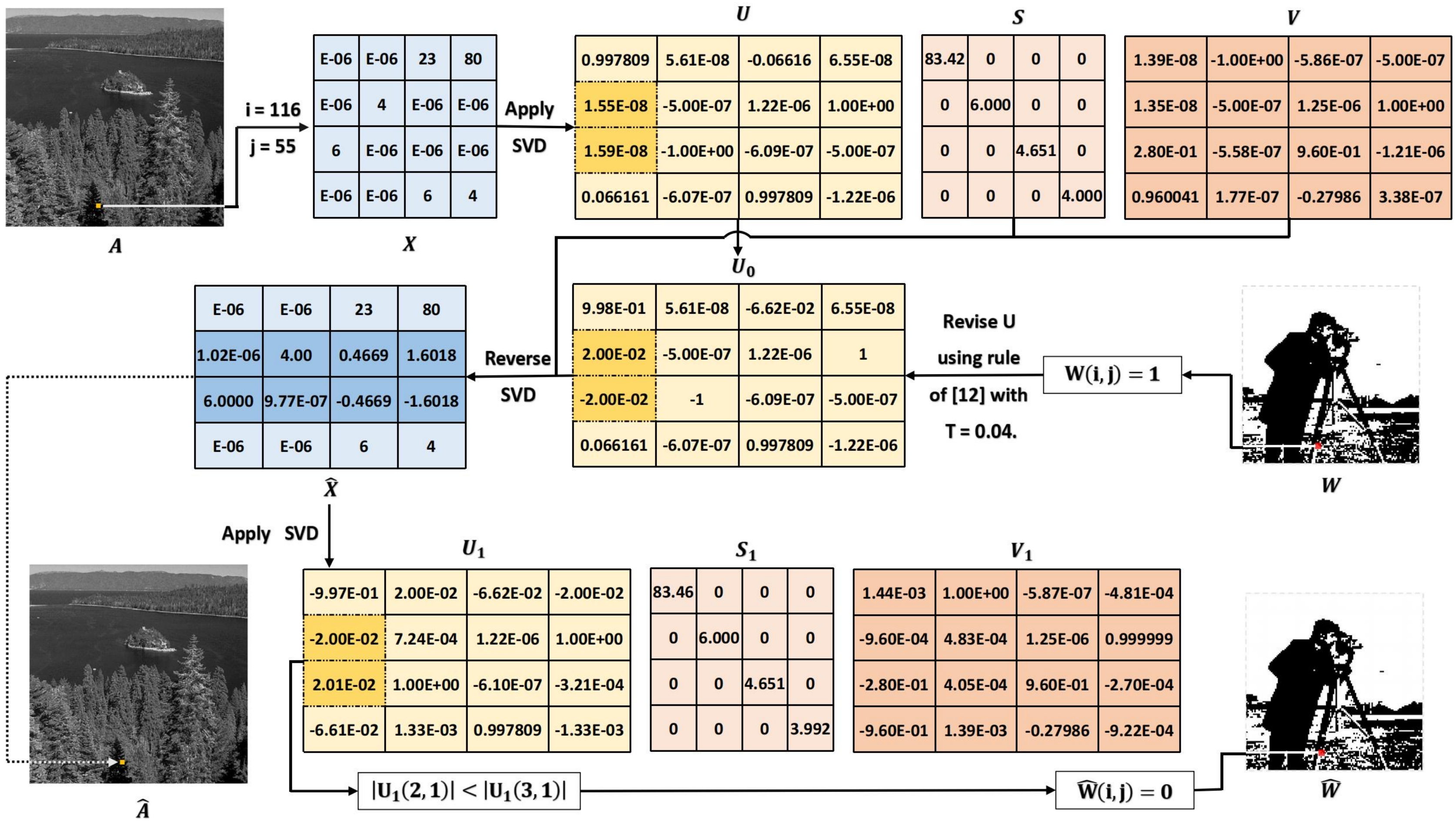


PSNR(I, Î)	
Mean	37.28 (dB)
Median	37.21 (dB)
Upper Quantile	40.45 (dB)
Lower Quantile	33.52 (dB)

[1] Wang H, Zhang Y, Zhao J. Enhancing the SVD compression losslessly[J]. Journal of Computational Science, 2023, 74: 102182.

4 The invertibility of embedding

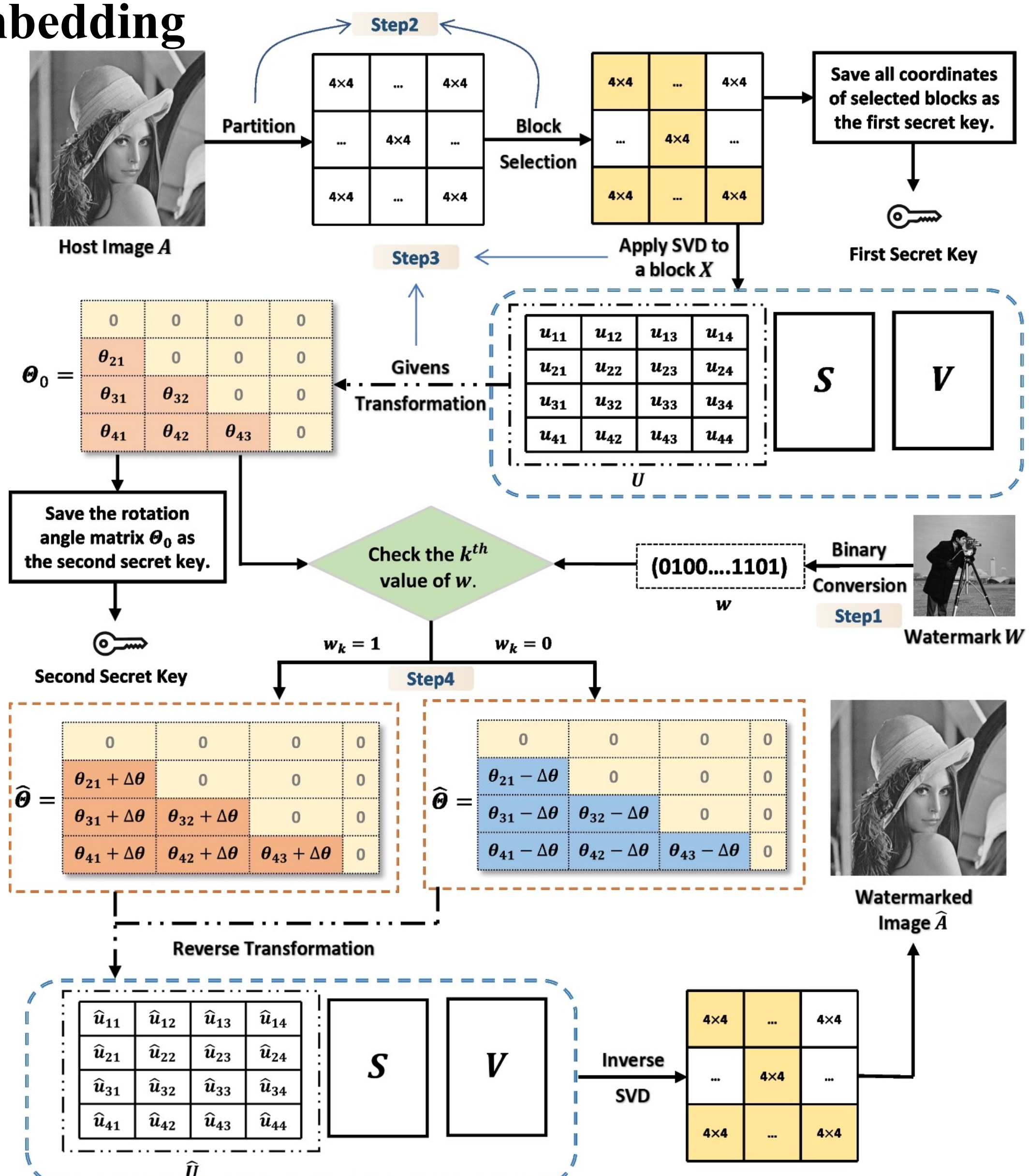
- Origin from a problem in digital watermarking:



[2] Zhang Y, Wang H, Zhao J. Eliminating orthonormal constraints of SVD to guarantee full retrievability of blind watermarking[J]. Multimedia Tools and Applications, 2023: 1-27.

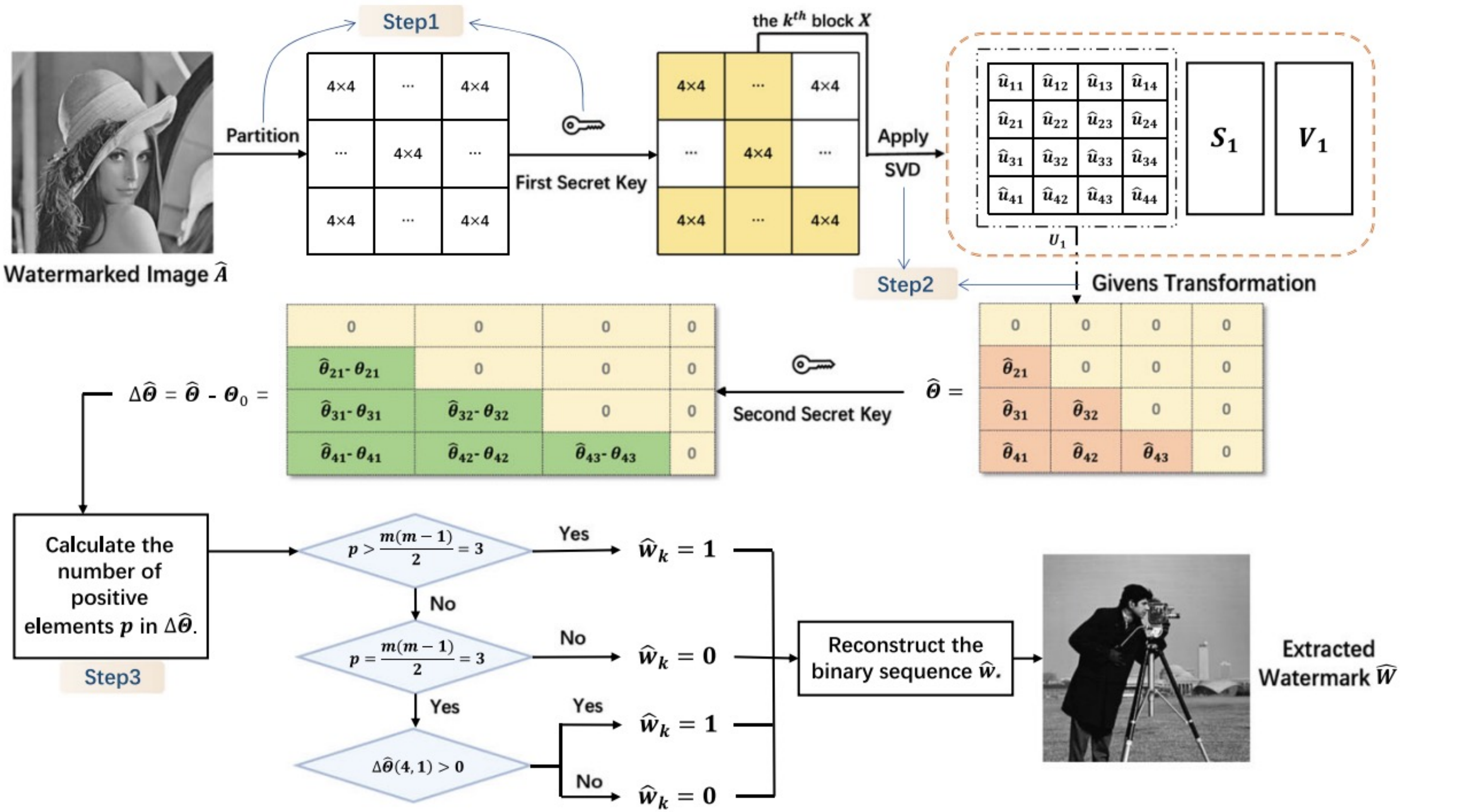
4 The invertibility of embedding

Watermark Embedding



4 The invertibility of embedding

Watermark Extraction

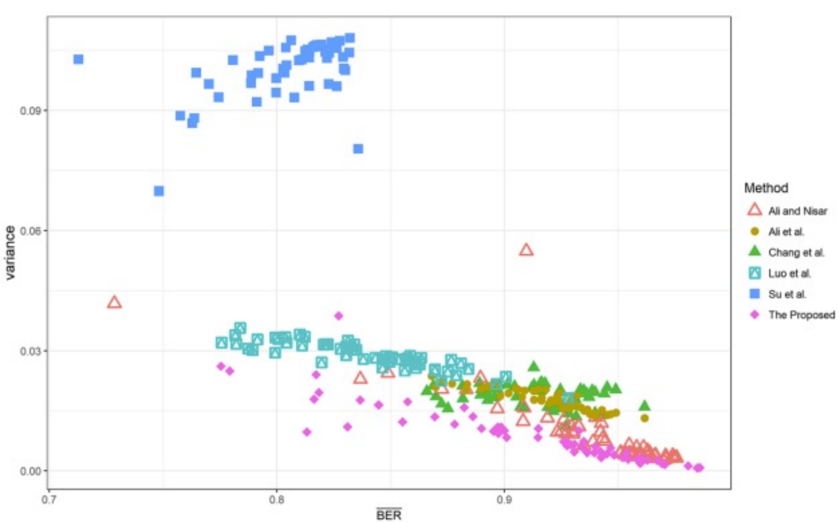


[2] Zhang Y, Wang H, Zhao J. Eliminating orthonormal constraints of SVD to guarantee full retrievability of blind watermarking[J]. Multimedia Tools and Applications, 2023: 1-27.

4 The invertibility of embedding

Experiment results – Robustness comparison

	Subbands	Modifying singular values	Modifying singular vectors
Ali et al.	LL of RIDWT		✓
Ali and Nisar	LL of DWT	✓	
Chang et al.	\		✓
Luo et al.	LL of DWT		✓
Su et al.	\	✓	
Proposed	\		✓



Attack type	Ali and Nisar	Ali et al.	Chang et al.	Luo et al.	Su et al.	Proposed
Gaussian Noise	0.9999	0.9999	0.9859	0.9991	0.9968	0.9958
S&P Noise	0.9998	0.9999	0.9957	0.9994	0.9967	0.9959
Speckle Noise	0.9999	1.0000	0.9957	0.9995	0.9967	0.9961
Average Filter	0.8858	0.7429	0.7377	0.6254	0.7482	0.7698
Gaussian Low-Pass Filter	0.9840	0.9624	0.9545	0.8381	0.9694	0.9475
JP2K CR=2	0.9949	0.9976	0.9926	0.9713	0.9968	0.9818
JP2K CR=4	0.9184	0.9620	0.9472	0.8373	0.9782	0.8828
JP2K CR=6	0.7827	0.8972	0.8768	0.7153	0.9266	0.7810
JPEG QF=50	0.7752	0.7423	0.7176	0.5726	0.9761	0.8259
JPEG QF=70	0.9041	0.8463	0.8315	0.6380	0.9932	0.8890
JPEG QF=90	0.9909	0.9771	0.9697	0.8309	0.9965	0.9574
Brighten	0.9996	0.9999	0.9958	0.9955	0.0198	0.9413
Darken	0.9043	0.9776	0.9556	0.9632	0.0224	0.9147

[2] Zhang Y, Wang H, Zhao J. Eliminating orthonormal constraints of SVD to guarantee full retrievability of blind watermarking[J]. Multimedia Tools and Applications, 2023: 1-27.

5 The sensitivity of embedding

- If the matrix is an image, are there any connection between the change of an image and the change of corresponding θ?

5 The sensitivity of embedding

- If the matrix is an image, are there any connection between the change of an image and the change of corresponding θ?

$$U = G_{21}^{T}G_{31}^{T}G_{41}^{T}G_{32}^{T}G_{42}^{T}G_{43}^{T}I_4$$

$$= \begin{bmatrix} cos\theta_{21} & -sin\theta_{21} & 0 & 0 \\ sin\theta_{21} & cos\theta_{21} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos\theta_{31} & 0 & -sin\theta_{31} & 0 \\ 0 & 1 & 0 & 0 \\ sin\theta_{31} & 0 & cos\theta_{31} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} cos\theta_{41} & 0 & 0 & -sin\theta_{41} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ sin\theta_{41} & 0 & 0 & cos\theta_{41} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\theta_{32} & -sin\theta_{32} & 0 \\ 0 & sin\theta_{32} & cos\theta_{32} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\theta_{42} & 0 & -sin\theta_{42} \\ 0 & 0 & 1 & 0 \\ 0 & sin\theta_{42} & 0 & cos\theta_{42} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & cos\theta_{43} & -sin\theta_{43} \\ 0 & 0 & sin\theta_{43} & cos\theta_{43} \end{bmatrix} I_4$$

$$= \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ u_{21} & u_{22} & u_{23} & u_{24} \\ u_{31} & u_{32} & u_{33} & u_{34} \\ u_{41} & u_{42} & u_{43} & u_{44} \end{bmatrix},$$

Recall our purpose that to detect image change quickly, it is reasonable to use the 1-rank approximation of $\boldsymbol{a}$ as the analysis object, then we can reconstruct $\boldsymbol{a}$ using θ.

$$\hat{a}_{11} = \lambda_1 (cos\theta_{21}cos\theta_{31}cos\theta_{41})^{\boldsymbol{U}} (cos\theta_{21}cos\theta_{31}cos\theta_{41})^{\boldsymbol{V}},$$

$$\hat{a}_{21} = \lambda_1 (sin\theta_{21}cos\theta_{31}cos\theta_{41})^{\boldsymbol{U}} (cos\theta_{21}cos\theta_{31}cos\theta_{41})^{\boldsymbol{V}},$$

$$\hat{a}_{31} = \lambda_1 (sin\theta_{31}cos\theta_{41})^{\boldsymbol{U}} (cos\theta_{21}cos\theta_{31}cos\theta_{41})^{\boldsymbol{V}},$$

$$\hat{a}_{41} = \lambda_1 (sin\theta_{41})^{\boldsymbol{U}} (cos\theta_{21}cos\theta_{31}cos\theta_{41})^{\boldsymbol{V}},$$

In order to figure out that when $\boldsymbol{a}$ changes, which θ would be more indicative, it is naturally for us to think about calculating derivatives.

$$\begin{aligned} d(\hat{a}_{41}) = & \lambda_1 (cos\theta_{41})^{\boldsymbol{U}} (cos\theta_{21}cos\theta_{31}cos\theta_{41})^{\boldsymbol{V}} \\ & \cdot d(\theta_{41}^{\boldsymbol{U}}) + \lambda_1 (sin\theta_{41})^{\boldsymbol{U}} (-sin\theta_{21}cos\theta_{31}cos\theta_{41})^{\boldsymbol{V}} \\ & \cdot d(\theta_{21}^{\boldsymbol{V}}) + \lambda_1 (sin\theta_{41})^{\boldsymbol{U}} (-cos\theta_{21}sin\theta_{31}cos\theta_{41})^{\boldsymbol{V}} \\ & \cdot d(\theta_{31}^{\boldsymbol{V}}) + \lambda_1 (sin\theta_{41})^{\boldsymbol{U}} (-cos\theta_{21}cos\theta_{31}sin\theta_{41})^{\boldsymbol{V}} \cdot d(\theta_{41}^{\boldsymbol{V}}) \end{aligned}$$

[3] Zhang Y, Zhao J. θ is all you need: Revisiting SVD in caputuring changes in matrices[C]//Proceedings of the 2022 5th International Conference on Algorithms, Computing and Artificial Intelligence. 2022: 1-9.

5 The sensitivity of embedding

- If the matrix is an image, are there any connection between the change of an image and the change of corresponding θ?

Specifically, we can find out for some elements, their total differentials are not depend on all angles, i.e. only relates to specific angles. Typically, elements in the forth line satisfy.

$$
\begin{aligned}
d(\hat{a}_{41}) =& \lambda_1 (cos\theta_{41})^{\boldsymbol{U}} (cos\theta_{21} cos\theta_{31} cos\theta_{41})^{\boldsymbol{V}} \\
& \cdot d(\theta_{41}^{\boldsymbol{U}}) + \lambda_1 (sin\theta_{41})^{\boldsymbol{U}} (-sin\theta_{21} cos\theta_{31} cos\theta_{41})^{\boldsymbol{V}} \\
& \cdot d(\theta_{21}^{\boldsymbol{V}}) + \lambda_1 (sin\theta_{41})^{\boldsymbol{U}} (-cos\theta_{21} sin\theta_{31} cos\theta_{41})^{\boldsymbol{V}} \\
& \cdot d(\theta_{31}^{\boldsymbol{V}}) + \lambda_1 (sin\theta_{41})^{\boldsymbol{U}} (-cos\theta_{21} cos\theta_{31} sin\theta_{41})^{\boldsymbol{V}} \cdot d(\theta_{41}^{\boldsymbol{V}}) \\
d(\hat{a}_{42}) =& \lambda_1 (cos\theta_{41})^{\boldsymbol{U}} (sin\theta_{21} cos\theta_{31} cos\theta_{41})^{\boldsymbol{V}} \\
& \cdot d(\theta_{41}^{\boldsymbol{U}}) + \lambda_1 (sin\theta_{41})^{\boldsymbol{U}} (cos\theta_{21} cos\theta_{31} cos\theta_{41})^{\boldsymbol{V}} \\
& \cdot d(\theta_{21}^{\boldsymbol{V}}) + \lambda_1 (sin\theta_{41})^{\boldsymbol{U}} (-sin\theta_{21} sin\theta_{31} cos\theta_{41})^{\boldsymbol{V}} \\
& \cdot d(\theta_{31}^{\boldsymbol{V}}) + \lambda_1 (sin\theta_{41})^{\boldsymbol{U}} (-sin\theta_{21} cos\theta_{31} sin\theta_{41})^{\boldsymbol{V}} \cdot d(\theta_{41}^{\boldsymbol{V}}) \\
d(\hat{a}_{43}) =& \lambda_1 (cos\theta_{41})^{\boldsymbol{U}} (sin\theta_{31} cos\theta_{41})^{\boldsymbol{V}} \cdot d(\theta_{41}^{\boldsymbol{U}}) \\
& + \lambda_1 (sin\theta_{41})^{\boldsymbol{U}} (cos\theta_{31} cos\theta_{41})^{\boldsymbol{V}} \cdot d(\theta_{31}^{\boldsymbol{V}}) \\
& + \lambda_1 (sin\theta_{41})^{\boldsymbol{U}} (-sin\theta_{31} sin\theta_{41})^{\boldsymbol{V}} \cdot d(\theta_{41}^{\boldsymbol{V}}) \\
d(\hat{a}_{44}) =& \lambda_1 (cos\theta_{41})^{\boldsymbol{U}} (sin\theta_{41})^{\boldsymbol{V}} \cdot d(\theta_{41}^{\boldsymbol{U}}) \\
& + \lambda_1 (sin\theta_{41})^{\boldsymbol{U}} (cos\theta_{41})^{\boldsymbol{V}} \cdot d(\theta_{41}^{\boldsymbol{V}}),
\end{aligned}
$$

5 The sensitivity of embedding

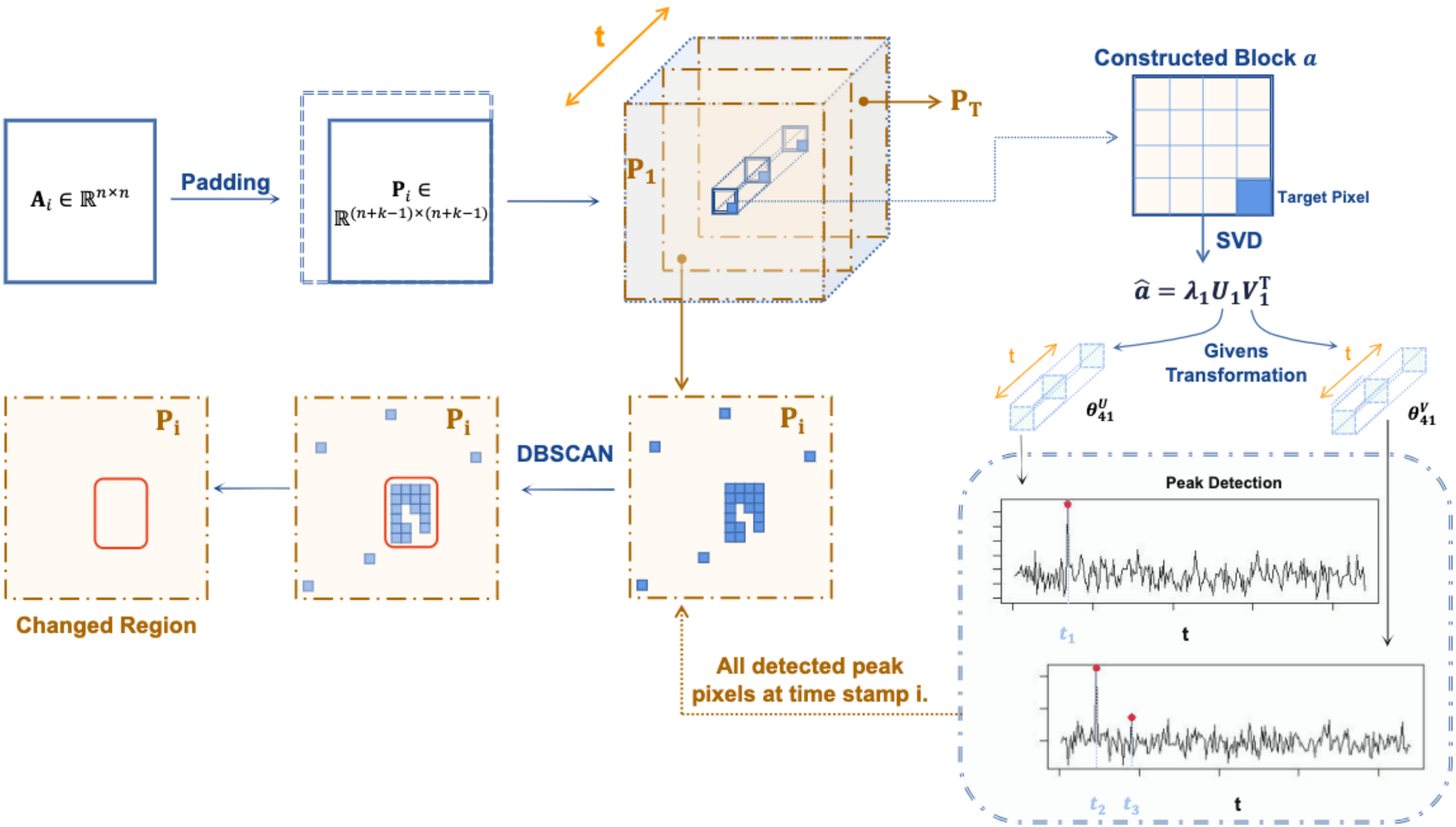
- If the matrix is an image, are there any connection between the change of an image and the change of corresponding θ?

$$
\begin{aligned}
d(\hat{a}_{41}) = & \lambda_1 (cos\theta_{41})^{\boldsymbol{U}} (cos\theta_{21} cos\theta_{31} cos\theta_{41})^{\boldsymbol{V}} \\
& \cdot d(\theta_{41}^{\boldsymbol{U}}) + \lambda_1 (sin\theta_{41})^{\boldsymbol{U}} (-sin\theta_{21} cos\theta_{31} cos\theta_{41})^{\boldsymbol{V}} \\
& \cdot d(\theta_{21}^{\boldsymbol{V}}) + \lambda_1 (sin\theta_{41})^{\boldsymbol{U}} (-cos\theta_{21} sin\theta_{31} cos\theta_{41})^{\boldsymbol{V}} \\
& \cdot d(\theta_{31}^{\boldsymbol{V}}) + \lambda_1 (sin\theta_{41})^{\boldsymbol{U}} (-cos\theta_{21} cos\theta_{31} sin\theta_{41})^{\boldsymbol{V}} \cdot d(\theta_{41}^{\boldsymbol{V}}) \\
d(\hat{a}_{42}) = & \lambda_1 (cos\theta_{41})^{\boldsymbol{U}} (sin\theta_{21} cos\theta_{31} cos\theta_{41})^{\boldsymbol{V}} \\
& \cdot d(\theta_{41}^{\boldsymbol{U}}) + \lambda_1 (sin\theta_{41})^{\boldsymbol{U}} (cos\theta_{21} cos\theta_{31} cos\theta_{41})^{\boldsymbol{V}} \\
& \cdot d(\theta_{21}^{\boldsymbol{V}}) + \lambda_1 (sin\theta_{41})^{\boldsymbol{U}} (-sin\theta_{21} sin\theta_{31} cos\theta_{41})^{\boldsymbol{V}} \\
& \cdot d(\theta_{31}^{\boldsymbol{V}}) + \lambda_1 (sin\theta_{41})^{\boldsymbol{U}} (-sin\theta_{21} cos\theta_{31} sin\theta_{41})^{\boldsymbol{V}} \cdot d(\theta_{41}^{\boldsymbol{V}}) \\
d(\hat{a}_{43}) = & \lambda_1 (cos\theta_{41})^{\boldsymbol{U}} (sin\theta_{31} cos\theta_{41})^{\boldsymbol{V}} \cdot d(\theta_{41}^{\boldsymbol{U}}) \\
& + \lambda_1 (sin\theta_{41})^{\boldsymbol{U}} (cos\theta_{31} cos\theta_{41})^{\boldsymbol{V}} \cdot d(\theta_{31}^{\boldsymbol{V}}) \\
& + \lambda_1 (sin\theta_{41})^{\boldsymbol{U}} (-sin\theta_{31} sin\theta_{41})^{\boldsymbol{V}} \cdot d(\theta_{41}^{\boldsymbol{V}}) \\
d(\hat{a}_{44}) = & \lambda_1 (cos\theta_{41})^{\boldsymbol{U}} (sin\theta_{41})^{\boldsymbol{V}} \cdot d(\theta_{41}^{\boldsymbol{U}}) \\
& + \lambda_1 (sin\theta_{41})^{\boldsymbol{U}} (cos\theta_{41})^{\boldsymbol{V}} \cdot d(\theta_{41}^{\boldsymbol{V}}),
\end{aligned}
$$

It is evident that, among all elements in $\widehat{\boldsymbol{a}}$, $\widehat{\boldsymbol{a}}^{44}$ is the most "independent" one, which means that its perturbation can be completely captured by only two θ, i.e., $\theta_{41}^{\boldsymbol{U}}$ and $\theta_{41}^{\boldsymbol{V}}$.

5 The sensitivity of embedding

- Proposed change detection scheme



[3] Zhang Y, Zhao J. θ is all you need: Revisiting SVD in caputuring changes in matrices[C]//Proceedings of the 2022 5th International Conference on Algorithms, Computing and Artificial Intelligence. 2022: 1-9.

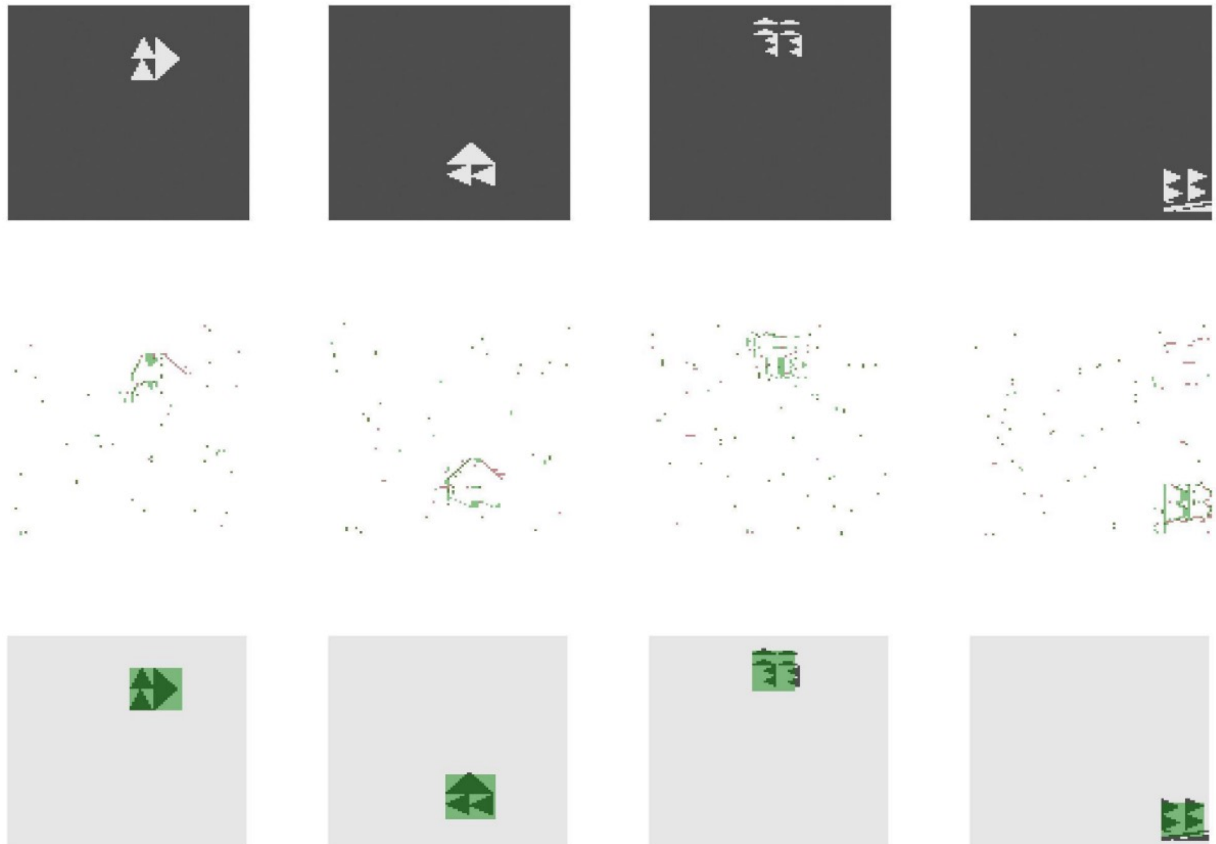
5 The sensitivity of embedding

- Simulation Result

Simulated Image

Detected changed pixels

Detected changed region

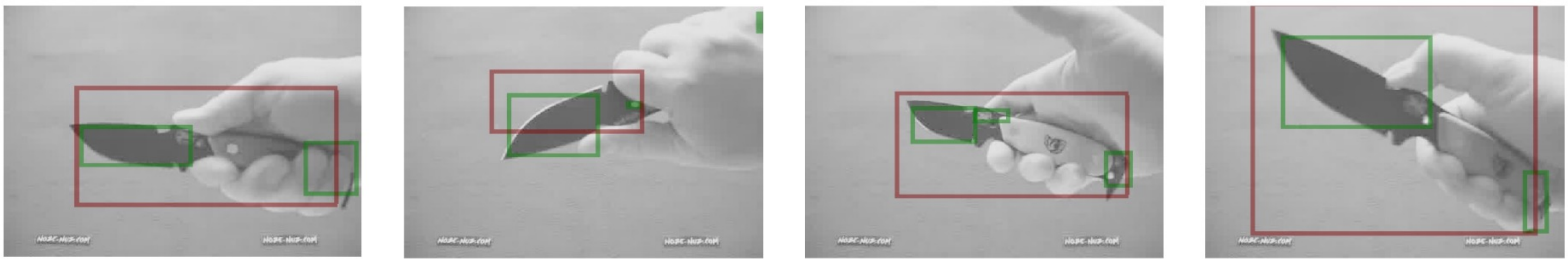


Median IOU	0.7960	0.8075	0.6192	0.6469

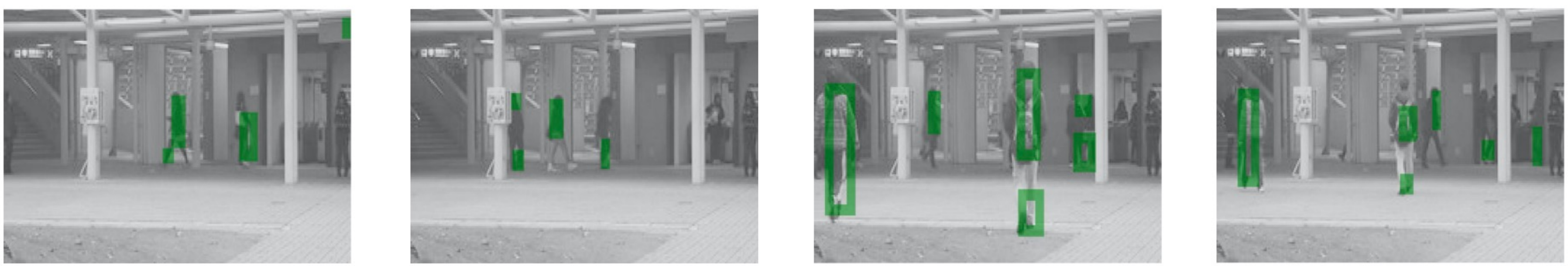
[3] Zhang Y, Zhao J. θ is all you need: Revisiting SVD in caputuring changes in matrices[C]//Proceedings of the 2022 5th International Conference on Algorithms, Computing and Artificial Intelligence. 2022: 1-9.

5 The sensitivity of embedding

- Real Result



(*a*) (*b*) (*c*) (*d*)



(*e*) (*f*) (*g*) (*h*)

[3] Zhang Y, Zhao J. θ is all you need: Revisiting SVD in caputuring changes in matrices[C]//Proceedings of the 2022 5th International Conference on Algorithms, Computing and Artificial Intelligence. 2022: 1-9.

6 Summary

Representation learning based on Givens transformation and its applications

- **The construction of embedding space**
  - **Theoretical Methodology**
  - **Compression Analysis**
- **The invertibility of embedding space**
  - **Real Problem in Digital Watermarking**
  - **Novel Watermarking Scheme**
- **The sensitivity of embedding space**
  - **Theoretical Deduction**
  - **Experimental results**

Thanks for your listening!