Robust Network Compressive Sensing

Yi-Chao Chen
The University of Texas at Austin

Joint work with
Lili Qiu[*], Yin Zhang[*], Guangtao Xue[+], Zhenxian Hu[+]
The University of Texas at Austin[*]
Shanghai Jiao Tong University[+]

ACM MobiCom 2014
September 10, 2014

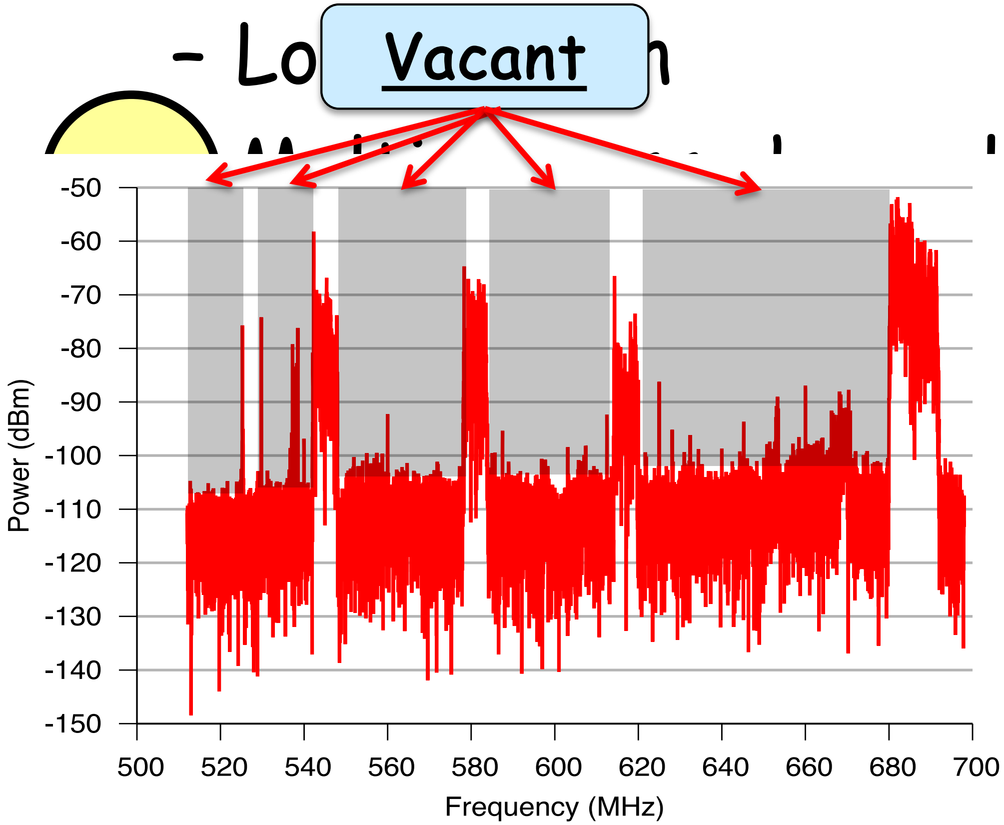
Network Matrices and Applications

- Network matrices
  - Traffic matrix
  - Loss matrix
  - Delay matrix
  - Channel State Information (CSI) matrix
  - RSS matrix

Q: How to fill in missing values in a matrix?

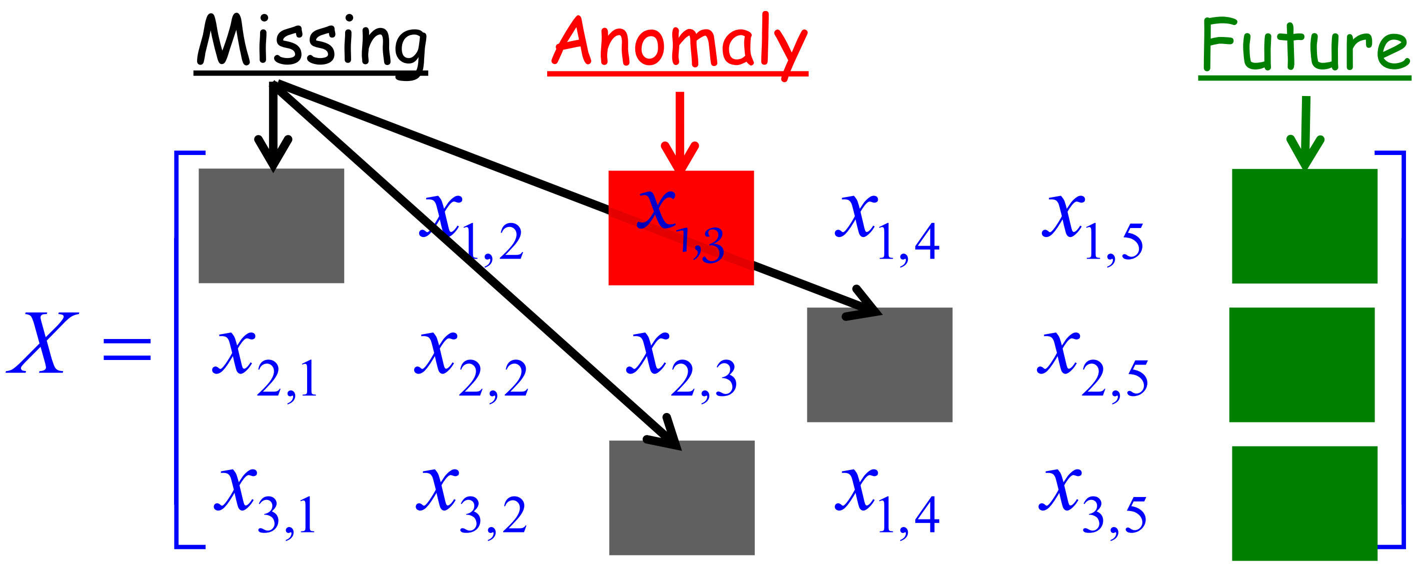
Missing Values: Why Bother?

- Applications need complete network matrices
  - Traffic engineering
  - Spectrum sensing
  - Channel estimation
  - Lo... n
  - ... design
  - ...less video coding



$$\begin{array}{c|cccc} & \text{time 1} & \text{time 2} & \cdots & \\ \text{freq 1} & x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} \\ \text{freq 2} & x_{2,1} & x_{2,2} & x_{2,3} & x_{2,4} \\ \text{freq 3} & x_{3,1} & x_{3,2} & x_{3,3} & x_{3,4} \end{array}$$

The Problem



Interpolation: fill in missing values from *incomplete*, *erroneous*, and/or *indirect* measurements

State of the Art

- Exploit low-rank nature of network matrices
  - matrices are low-rank [LPCD+04, LCD04, SIGCOMM09]:

    $\mathbf{X}_{nxm} \approx \mathbf{L}_{nxr} * \mathbf{R}_{mxr}^T \quad (r \ll n,m)$

- Exploit spatio-temporal properties
  - matrix rows or columns close to each other are often close in value [SIGCOMM09]

- Exploit local structures in network matrices
  - matrices have both global & local structures
  - Apply K-Nearest Neighbor (KNN) for local interpolation [SIGCOMM09]

Limitation

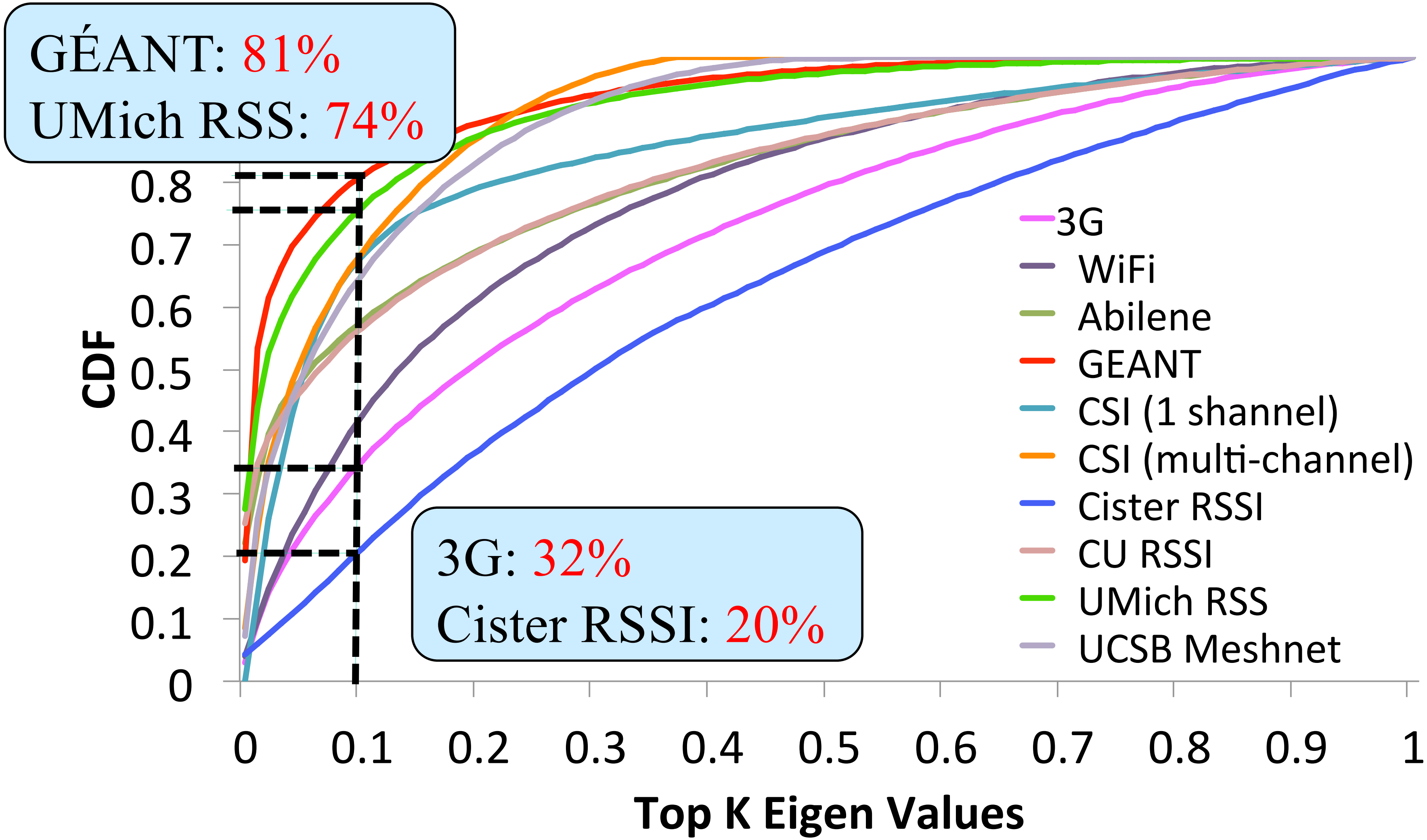
- Many factors contribute to network matrices
  - Anomalies, measurement errors, and noise
  - These factors may destroy low-rank structure and spatio-temporal locality

Network Matrices

Network	Date	Duration	Size (flows/links x #timeslot)
3G traffic	11/2010	1 day	472 x 144
WiFi traffic	1/2013	1 day	50 x 118
Abilene traffic	4/2003	1 week	121 x 1008
GEANT traffic	4/2005	1 week	529 x 672
1 channel CSI	2/2009	15 min.	90 x 9000
Multi. Channel CSI	2/2014	15 min.	270 x 5000
Cister RSSI	11/2010	4 hours	16 x 10000
CU RSSI	8/2007	500 frames	895 x 500
Umich RSS	4/2006	30 min.	182 x 3127
UCSB Meshnet	4/2006	3 days	425 x 1527

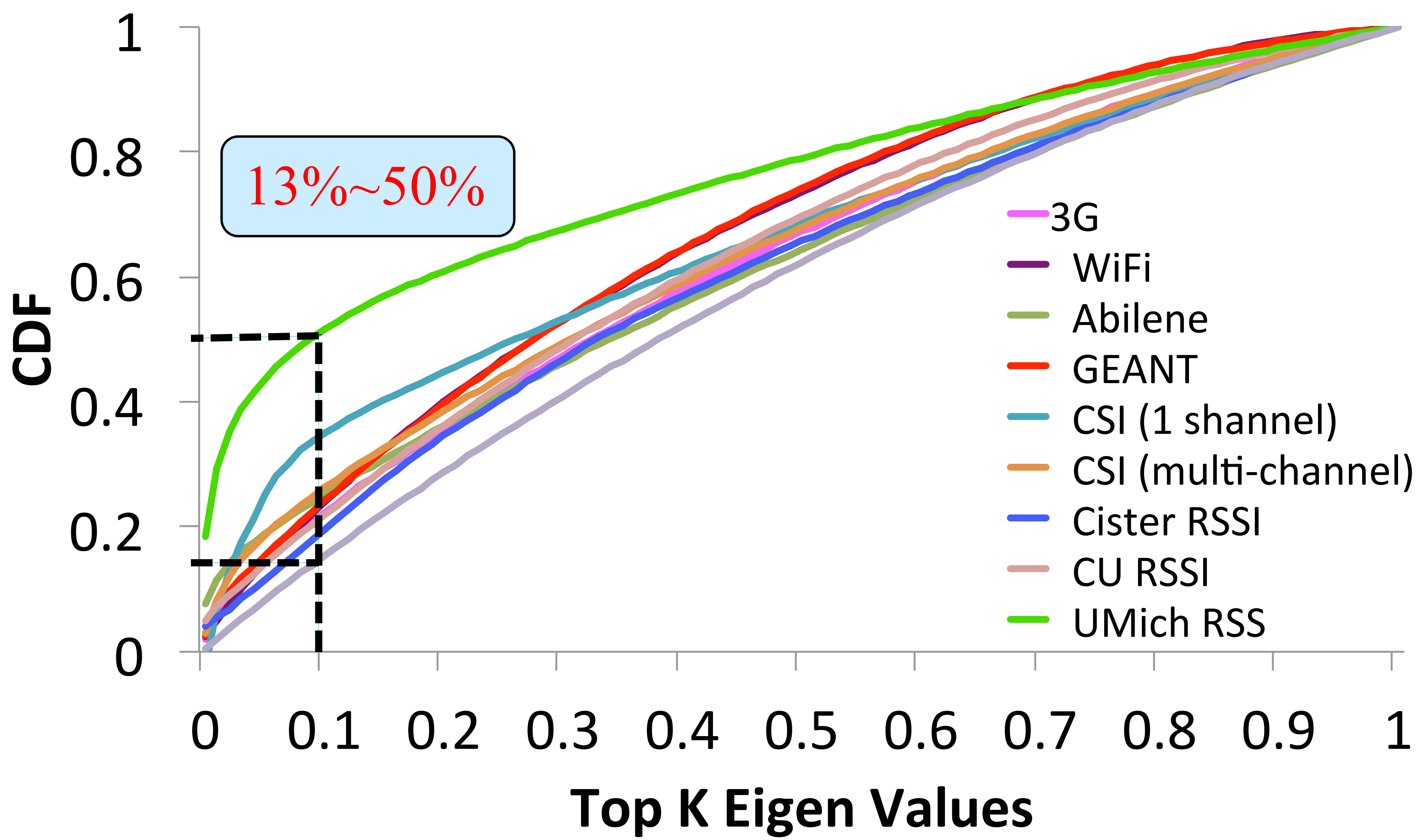
Rank Analysis

Not all matrices are low rank.



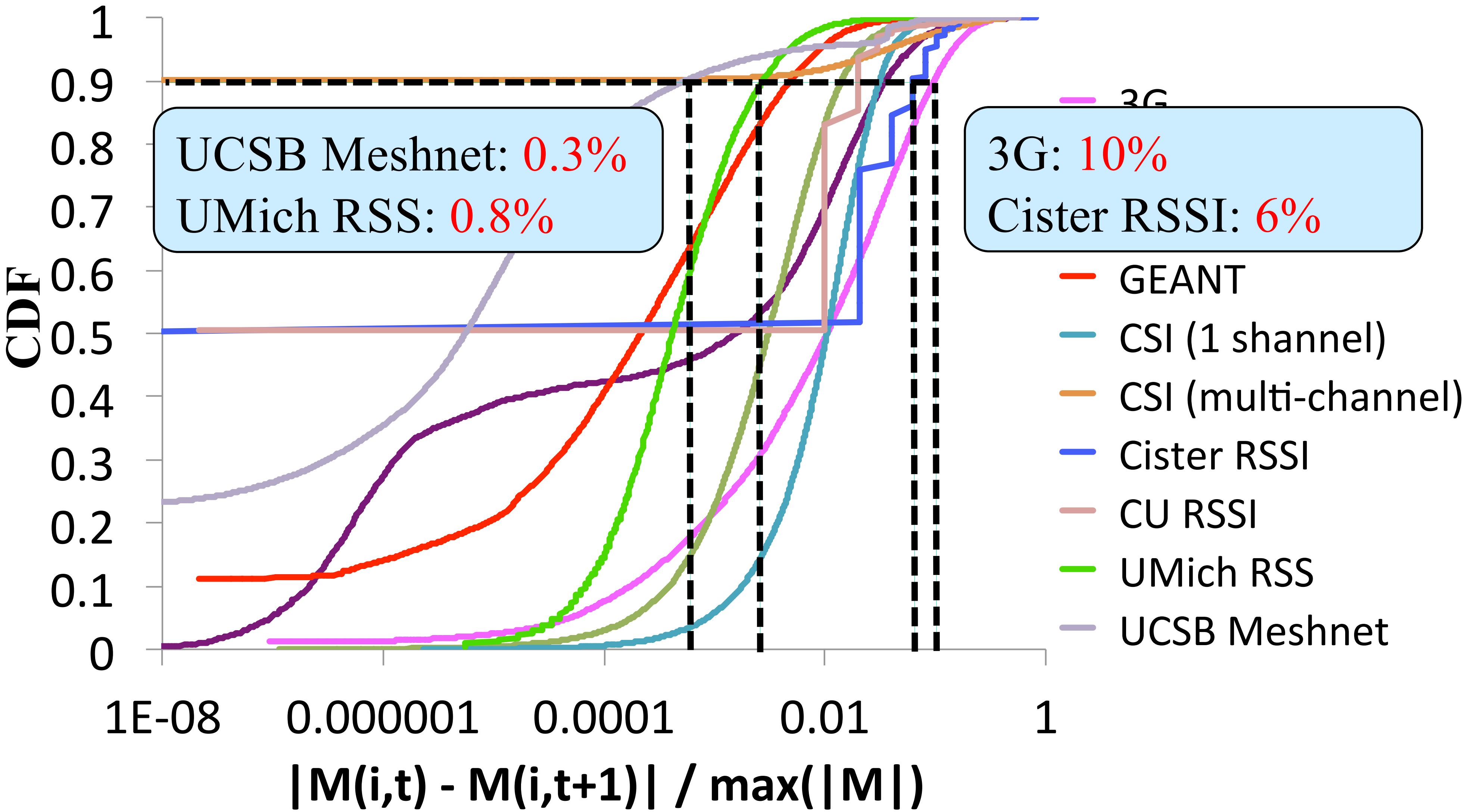
Rank Analysis (Cont.)

Adding anomalies increases rank in all traces



Temporal Stability

Temporal stability varies across traces.



Summary of the Analyses

- Our analyses reveal
  - Real network matrices may not be low rank
  - Adding anomalies increases the rank
  - Temporal stability varies across traces

Challenges

- How to explicitly account for anomalies, errors, and noise ?
- How to support matrices with different temporal stability?

Robust Compressive Sensing

- A new matrix decomposition that is general and robust against error/anomalies
  - Low rank matrix, anomaly matrix, noise matrix
- A self-learning algorithm to automatically tune the parameters
  - Account for varying temporal stability
- An efficient optimization algorithm
  - Search for the best parameters
  - Work for large network matrices

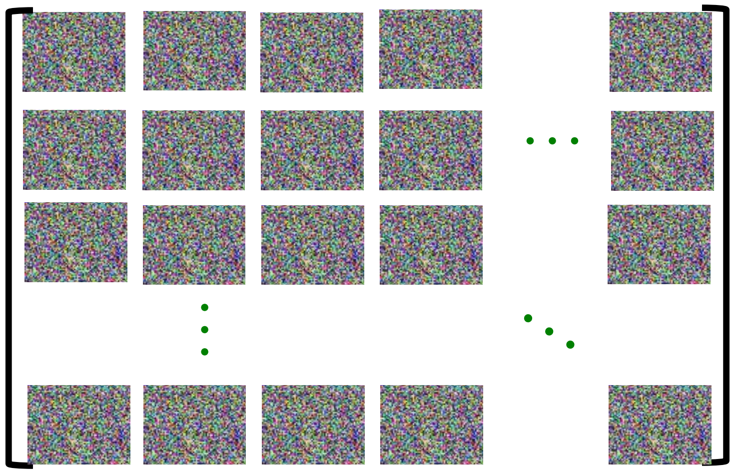
LENS Decomposition: Basic Formulation

$$
\begin{bmatrix}
\blacksquare & d_{1,2} & d_{1,3} & d_{1,4} & & d_{1,n} \\
d_{2,1} & d_{2,2} & d_{2,3} & \blacksquare & \cdots & d_{2,n} \\
d_{3,1} & d_{3,2} & \blacksquare & d_{3,4} & & d_{3,n} \\
 & \vdots & & & \ddots & \\
\blacksquare & d_{m,2} & d_{m,3} & d_{m,4} & & d_{m,n}
\end{bmatrix}
=
\begin{bmatrix}
x_{1,1} & & x_{1,r} \\
x_{2,1} & & x_{2,r} \\
x_{3,1} & \cdots & x_{3,r} \\
 & \vdots & \\
x_{m,1} & & x_{m,r}
\end{bmatrix}
\begin{bmatrix}
x_{1,1} & x_{1,2} & x_{1,3} & & x_{1,n} \\
 & & \vdots & \cdots & \\
x_{r,1} & x_{r,2} & x_{r,3} & & x_{r,n}
\end{bmatrix}
+
$$

[Input] D:
Original matrix

[Output] X:
A low rank matrix (r « m,n)

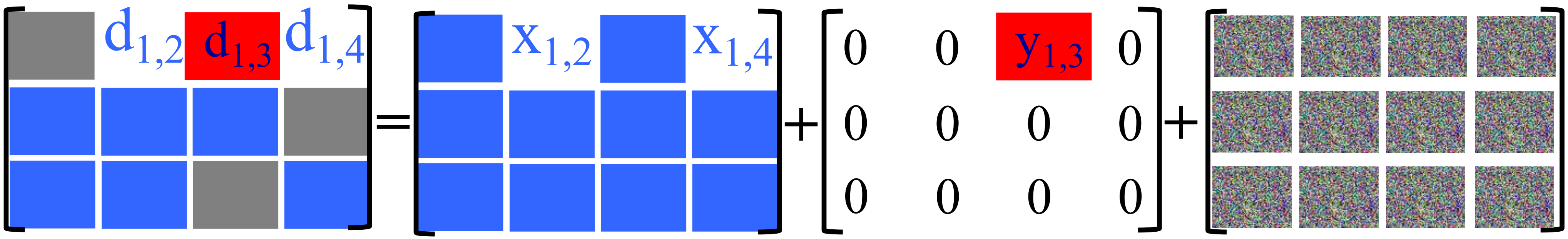
$$
\begin{bmatrix}
0 & 0 & y_{1,3} & 0 & & 0 \\
0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & & y_{3,n} \\
 & \vdots & & & \ddots & \\
0 & 0 & 0 & 0 & & 0
\end{bmatrix}
+
$$



[Output] Y:
A sparse anomaly matrix

[Output] Z:
A small noise matrix

LENS Decomposition: Basic Formulation



[Input] D: Original matrix

[Output] X: A low rank matrix

[Output] Y: A sparse anomaly matrix

[Output] Z: A small noise matrix

- Formulate it as a convex opt. problem:

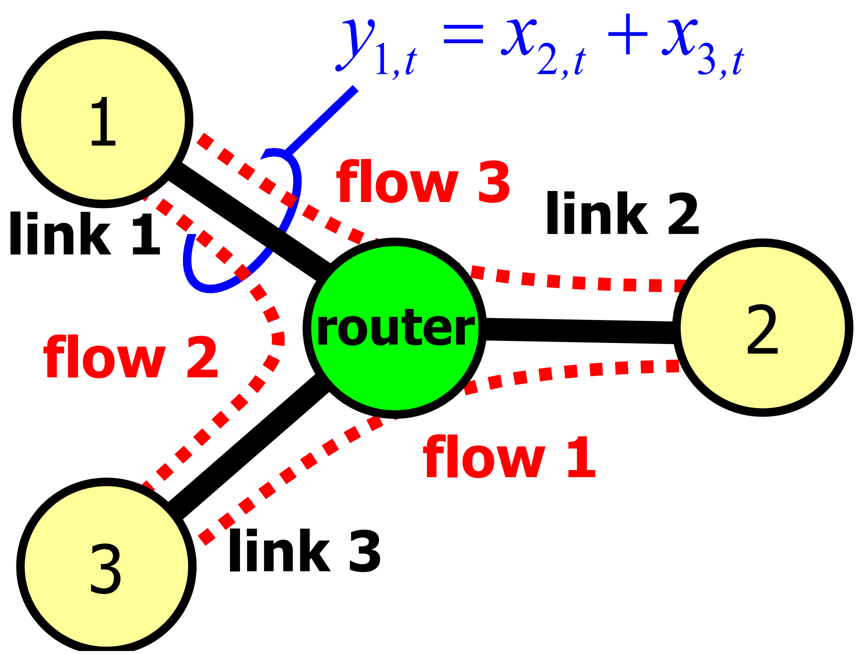
$$\textit{min:} \quad \alpha \boxed{\|X\|_*} + \beta \boxed{\|Y\|_1} + \frac{1}{2\sigma} \boxed{\|Z\|_F^2}$$

$$\textit{subject to:} \quad X + Y + Z + W = D$$

$$E.*W = W$$

LENS Decomposition: Support Indirect Measurement

- The matrix of interest may not be directly observable (e.g., traffic matrices)
  - AX + BY + CZ + W = D
    - A: routing matrix
    - B: an over-complete anomaly profile matrix
    - C: noise profile matrix



LENS Decomposition: Account for Domain Knowledge

- Domain Knowledge
  - Temporal stability
  - Spatial locality
  - Initial solution

$$\frac{\gamma}{2\sigma}\left\| X \begin{bmatrix} 1 & -1 & 0 & \cdots \\ 0 & 1 & -1 & \ddots \\ 0 & 0 & 1 & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix}^T \right\|_F^2$$

$$min: \ \alpha\|X\|_* + \beta\|Y\|_1 + \frac{1}{2\sigma}\|Z\|_F^2 + \frac{\gamma}{2\sigma}\sum_{k=1}^{K}\left\|P_k X Q_k^T - R_k\right\|_F^2$$

$$subject\ to: \ AX + BY + CZ + W = D$$

$$E.*W = W$$

Optimization Algorithm

- One of many challenges in optimization:
  - X and Y appear in multiple places in the objective and constraints
  - Coupling makes optimization hard
- Reformulation for optimization by introducing auxiliary variables
- $$min:\ \alpha\|X\|_* + \beta\|Y\|_1 + \frac{1}{2\sigma}\|Z\|_F^2 + \frac{\gamma}{2\sigma}\sum_{k=1}^{K}\|P_k X_k Q_k^T - R_k\|_F^2$$

$$\begin{aligned} subject\ to:\ & AX_0 + BY_0 + CZ + W = D \\ & E.*W = W \\ & X_k - X = 0 \quad (\forall k = 0,\ldots,K) \\ & Y_0 - Y = 0 \end{aligned}$$

Optimization Algorithm

- Alternating Direction Method (ADM)
  - For each iteration, alternate among the optimization of the augmented Lagrangian function by varying each one of X, X_k, Y, Y_0, Z, W, M, M_k, N while fixing the other variables
  - Improve efficiency through approximate SVD

Setting Parameters

$$min:\ \boldsymbol{\alpha}\|X\|_* + \boldsymbol{\beta}\|Y\|_1 + \frac{1}{2\boldsymbol{\sigma}}\|Z\|_F^2 + \frac{\gamma}{2\sigma}\sum_{k=1}^{K}\left\|P_k X_k Q_k^T - R_k\right\|_F^2$$

- $\alpha = \left(\sqrt{m_x} + \sqrt{n_x}\right)\left(\sqrt{\eta(D)}\right)$
- $\beta = \sqrt{2\log\left(m_Y n_Y\right)}$
- $\sigma = \theta\sigma_D$

where (m_X,n_X) is the size of X,
(m_Y,n_Y) is the size of Y,
η(D) is the fraction of entries neither missing or erroneous,
θ is a control parameter that limits contamination of dense measurement noise

Setting Parameters (Cont.)

$$min\!: \ \alpha\|X\|_* + \beta\|Y\|_1 + \frac{1}{2\sigma}\|Z\|_F^2 + \frac{\gamma}{2\sigma}\sum_{k=1}^{K}\|P_k X_k Q_k^T - R_k\|_F^2$$

- ϒ reflects the importance of domain knowledge
  - e.g. temporal-stability varies across traces
- Self-tuning algorithm
  - Drop additional entries in the matrix
  - Quantify the error of the entries that were present in the matrix but dropped intentionally during the search
  - Pick ϒ that gives lowest error

Evaluation Methodology

- Metric
  - Normalized Mean Absolute Error for missing values

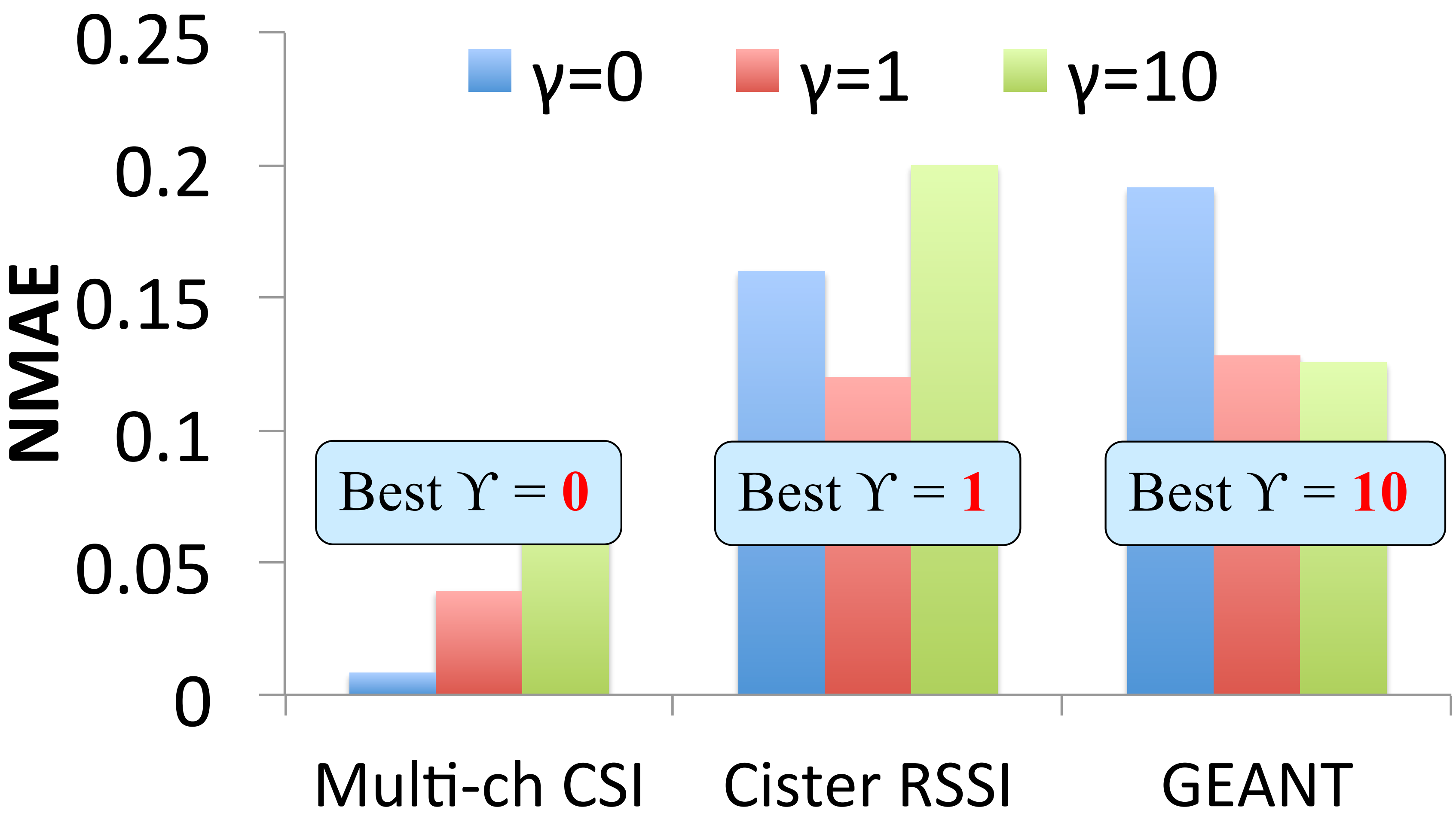
$$NMAE = \frac{\sum_{i,j:M(i,j)=0} \left| X(i,j) - X_{est}(i,j) \right|}{\sum_{i,j:M(i,j)=0} \left| X(i,j) \right|}$$

- Report the average of 10 random runs
- Anomaly generation
  - Inject anomalies to a varying fraction of entries with varying sizes
- Different dropping models

Algorithms Compared

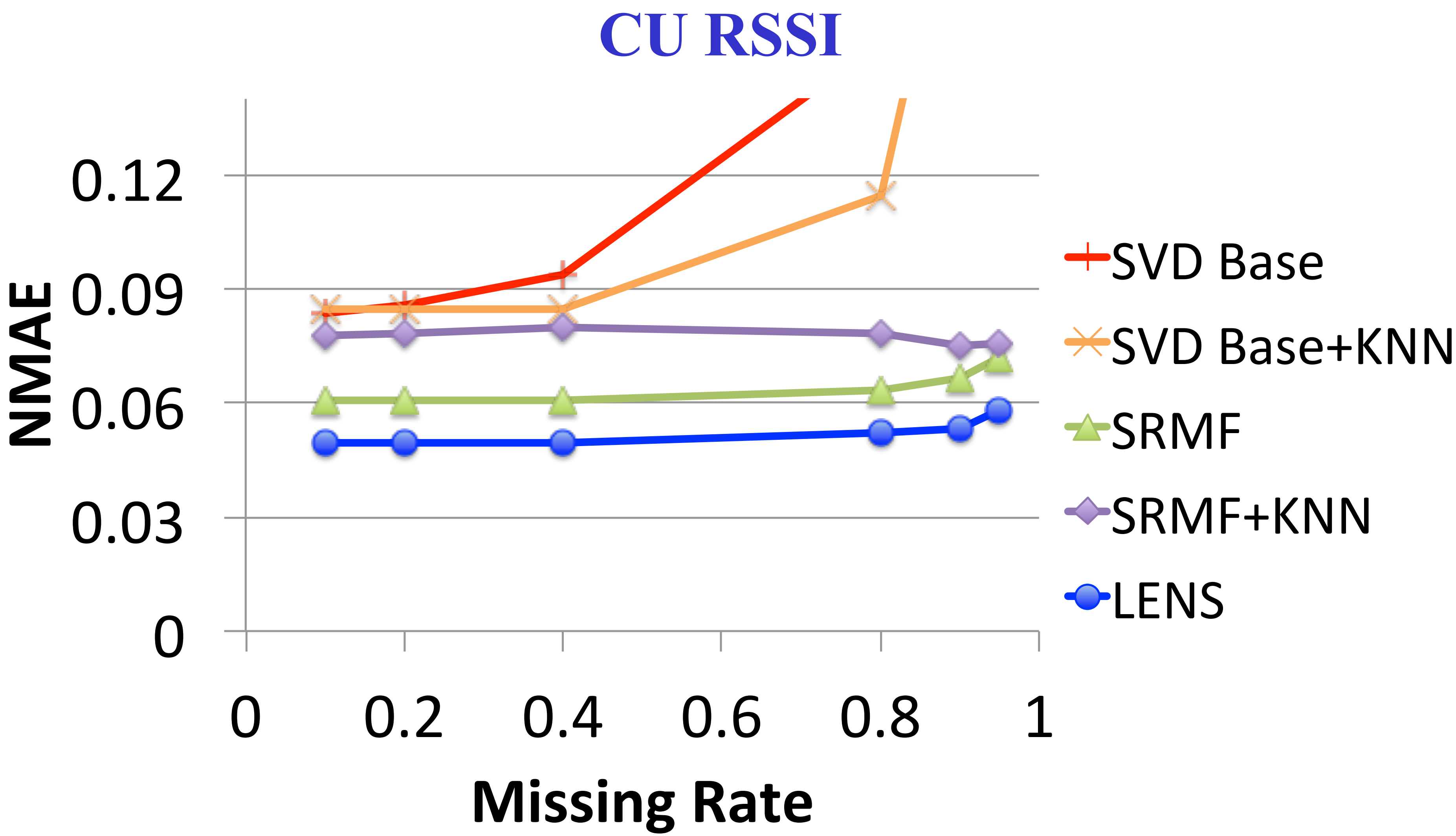
Algorithm	Description
Baseline	Baseline estimate via rank-2 approximation
SVD-base	SRSVD with baseline removal
SVD-base +KNN	Apply KNN after SVD-base
SRMF [SIGCOMM09]	Sparsity Regularized Matrix Factorization
SRMF+KNN [SIGCOMM09]	Hybrid of SRMF and KNN
LENS	Robust network compressive sensing

Self Learned ϒ



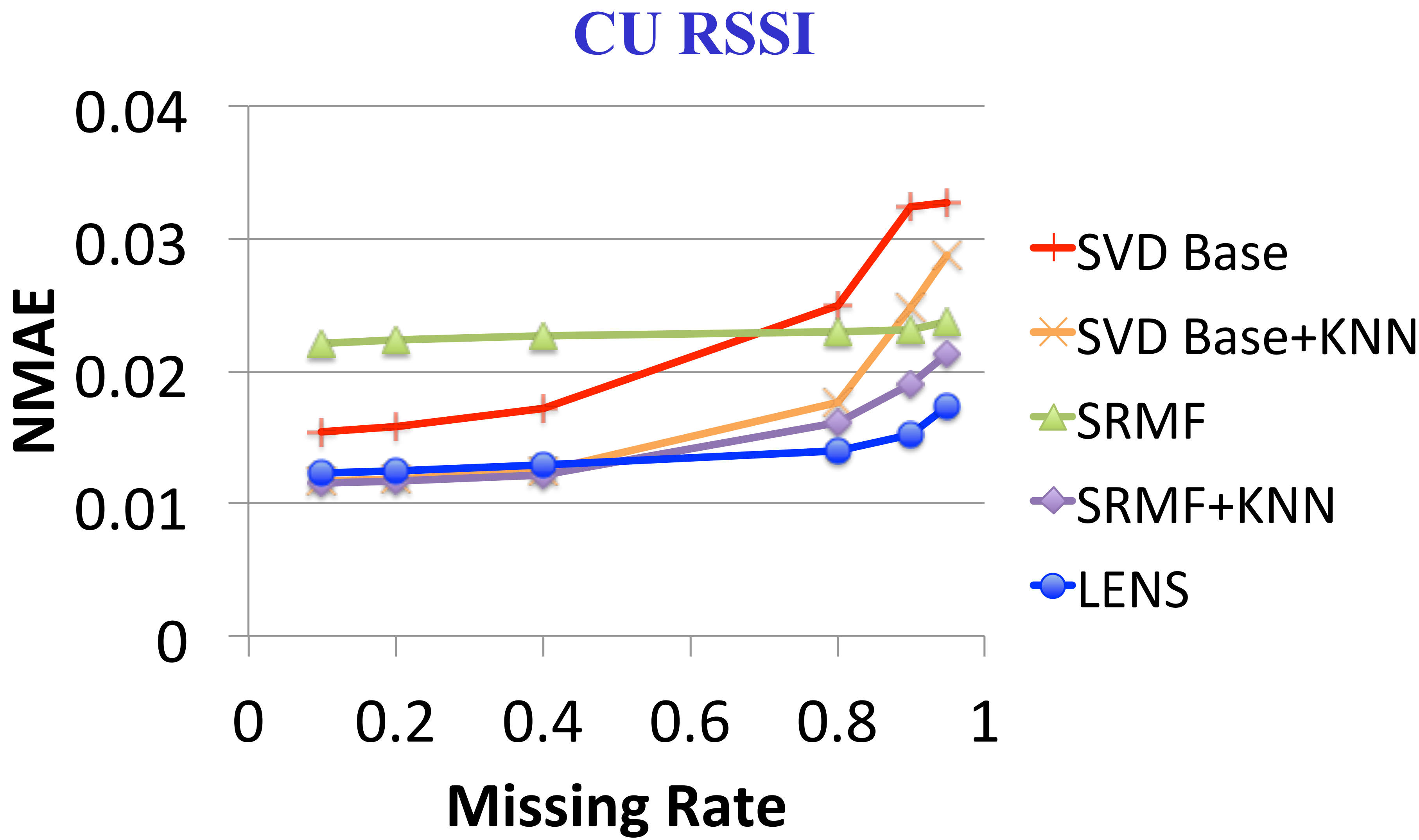
No single ϒ works for all traces.
Self tuning allows us to automatically select the best ϒ.

Interpolation under anomalies



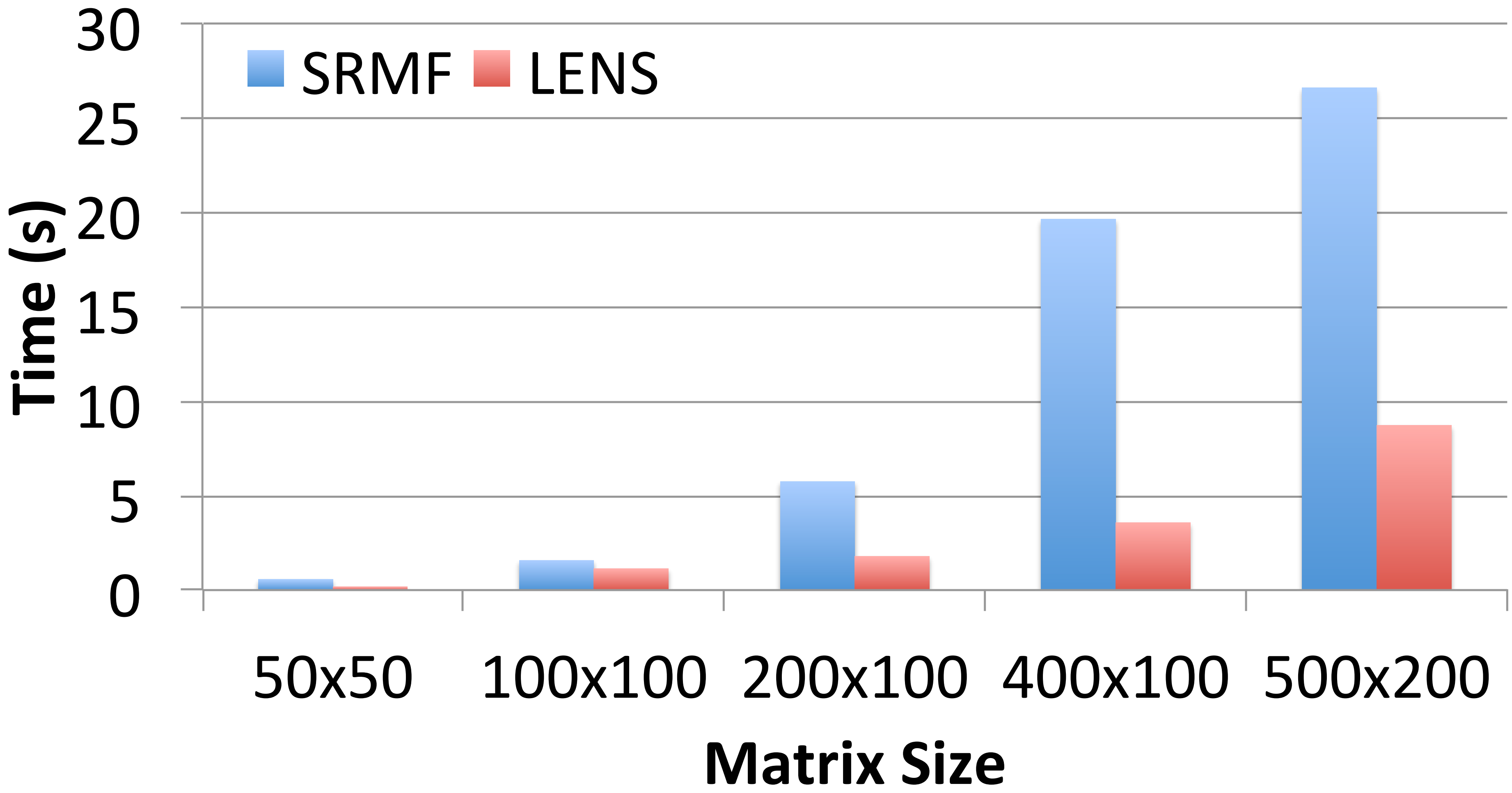
LENS performs the best under anomalies.

Interpolation without anomalies



LENS performs the best even without anomalies.

Computation Time



LENS is efficient due to local optimization in ADM.

Summary of Other Results

- The improvement of LENS increases with anomaly sizes and # anomalies.
- LENS consistently performs the best under different dropping modes.
- LENS yields the lowest prediction error.
- LENS achieves higher anomaly detection accuracy.

Conclusion

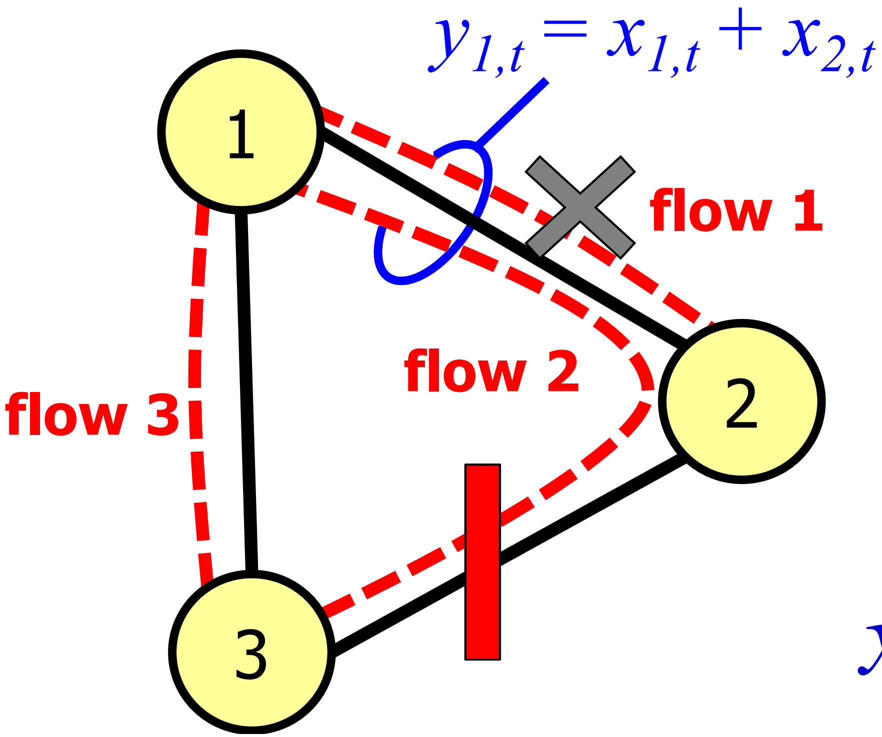
- Main contributions
  - Important impact of anomalies in matrix interpolation
  - Decompose a matrix into
    - a low-rank matrix,
    - a sparse anomaly matrix,
    - a dense but small noise matrix
  - An efficient optimization algorithm
  - A self-learning algorithm to automatically tune the parameters
- Future work
  - Applying it to spectrum sensing, channel estimation, localization, etc.

Thank you!

BACKUP SLIDES

Missing Values: Why Bother?

- Missing values are common in network matrices
  - Measurement and data collection are unreliable
  - Anomalies/outliers hide non-anomaly-related traffic
  - Future entries has not yet appeared
  - Direct measurement is infeasible/expensive



$$
\begin{array}{ccccc}
 & \text{Missing} & & \text{Anomaly} & \text{Future} \\
\end{array}
$$

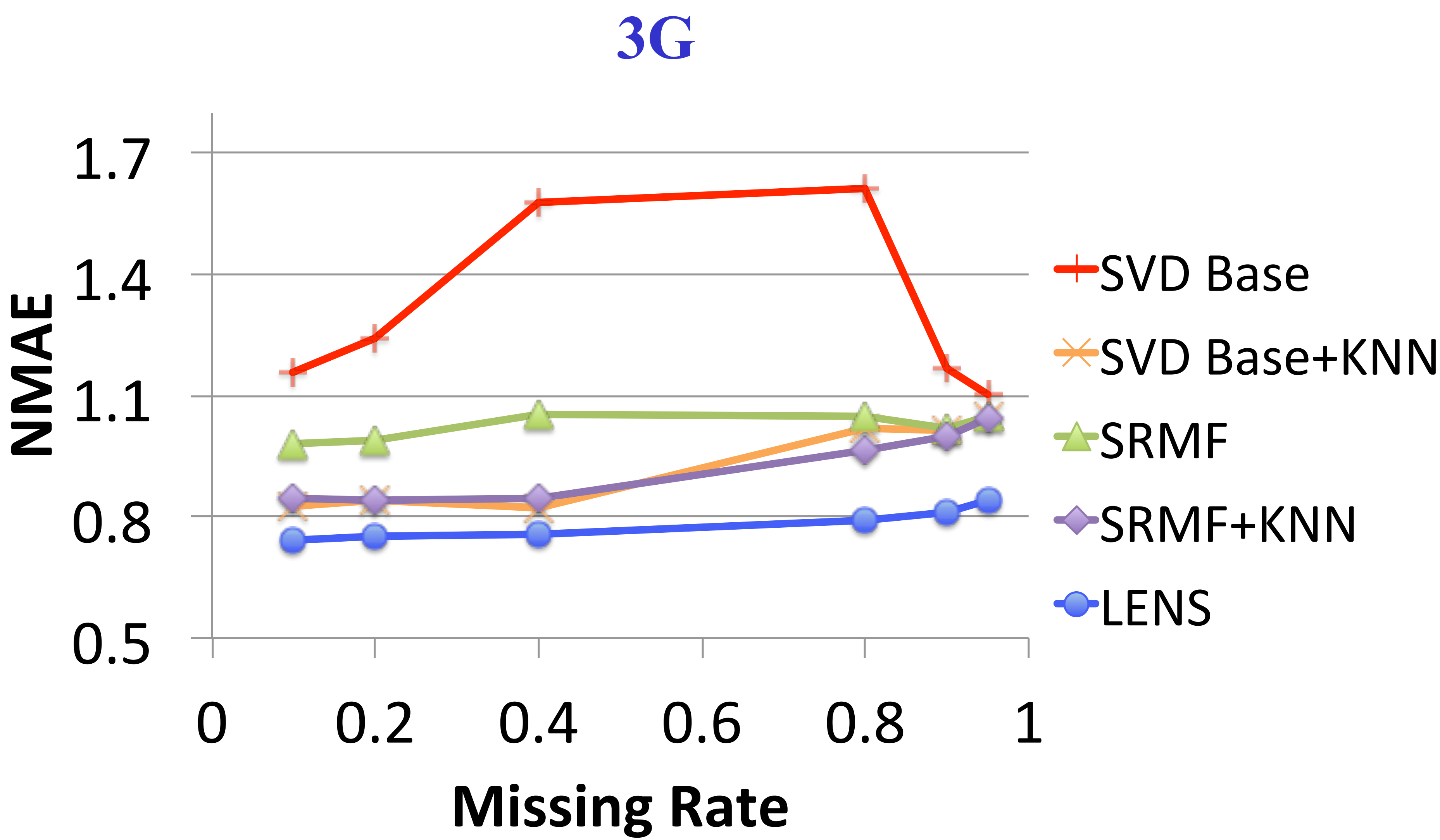
$$
\begin{bmatrix}
x_{1,1} & ? & ? & x_{1,4} & ? \\
x_{2,1} & x_{2,2} & x_{2,3} & x_{2,4} & ? \\
x_{3,1} & x_{3,2} & x_{3,3} & x_{3,4} & ?
\end{bmatrix}
$$

$x_{f,t}$: traffic of flow f at time t

Anomaly Generation

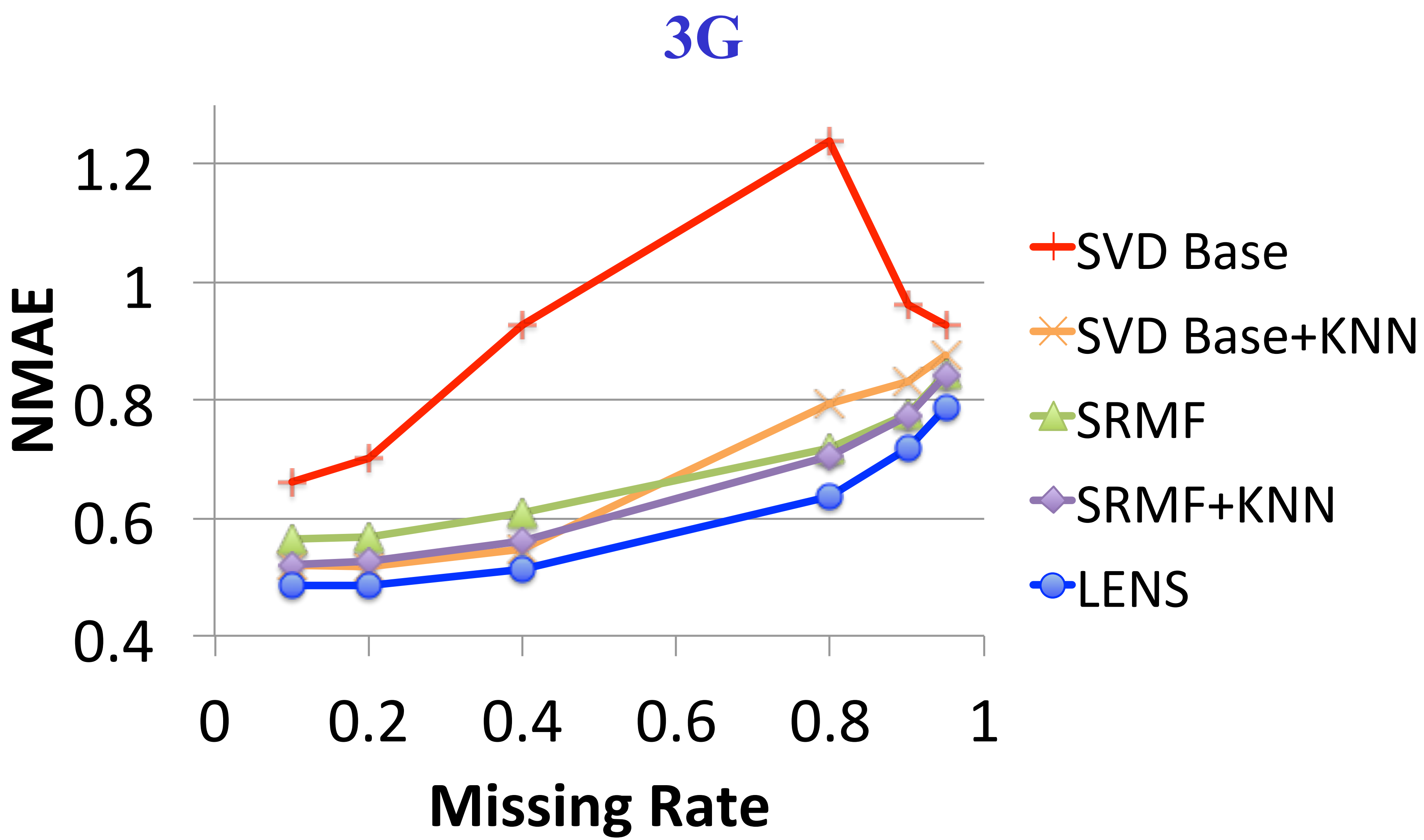
- Anomalies in real-world network matrices
  - The ground truth is hard to get
  - Injecting the same size of anomalies for all network matrices is not practical
- Generating anomalies based on the nature of the network matrix [SIGCOMM04, INFOCOM07, PETS11]
  - Apply Exponential Weighted Moving Average (EWMA) to predict the matrix
  - Calculate the difference between the real matrix and the predicted matrix
    - The difference is due to prediction error or anomalies
  - Sort the difference and select the largest one as the size of anomalies

Interpolation under anomalies



LENS performs the best under anomalies.

Interpolation without anomalies



LENS performs the best even without anomalies.

Non-monotonicity of Performance

- SVD base method is more sensitive the anomalies
- performance is affected by both the missing rate and the number of anomalies.
  - As the missing rate increases, the number of anomalies reduces.
    - When the missing rate increase, error increases
    - When the anomalies reduces, error reduces

Dropping Models

- Pure Random Loss
  - Elements are dropped independently with a random loss rate

$$
\begin{bmatrix}
d_{1,1} & \blacksquare & d_{1,3} & d_{1,4} & & d_{1,n} \\
d_{2,1} & d_{2,2} & d_{2,3} & d_{2,4} & \cdots & d_{2,n} \\
d_{3,1} & d_{3,2} & \blacksquare & d_{3,4} & & \blacksquare \\
 & \vdots & & & \ddots & \\
\blacksquare & d_{m,2} & d_{m,3} & d_{m,4} & & d_{m,n}
\end{bmatrix}
$$

Dropping Models

- Time Rand Loss
  - Columns are dropped
  - To emulate random losses during certain times
    - e.g. disk becomes full

$$\begin{bmatrix} d_{1,1} & & & d_{1,4} & & d_{1,n} \\ d_{2,1} & & & d_{2,4} & \cdots & d_{2,n} \\ d_{3,1} & & & d_{3,4} & & d_{3,n} \\ & & & & \ddots & \\ d_{m,1} & & & d_{m,4} & & d_{m,n} \end{bmatrix}$$

Dropping Models

- Element Rand Loss
  - Rows are dropped
  - To emulate certain nodes lose data
    - e.g., due to battery drain

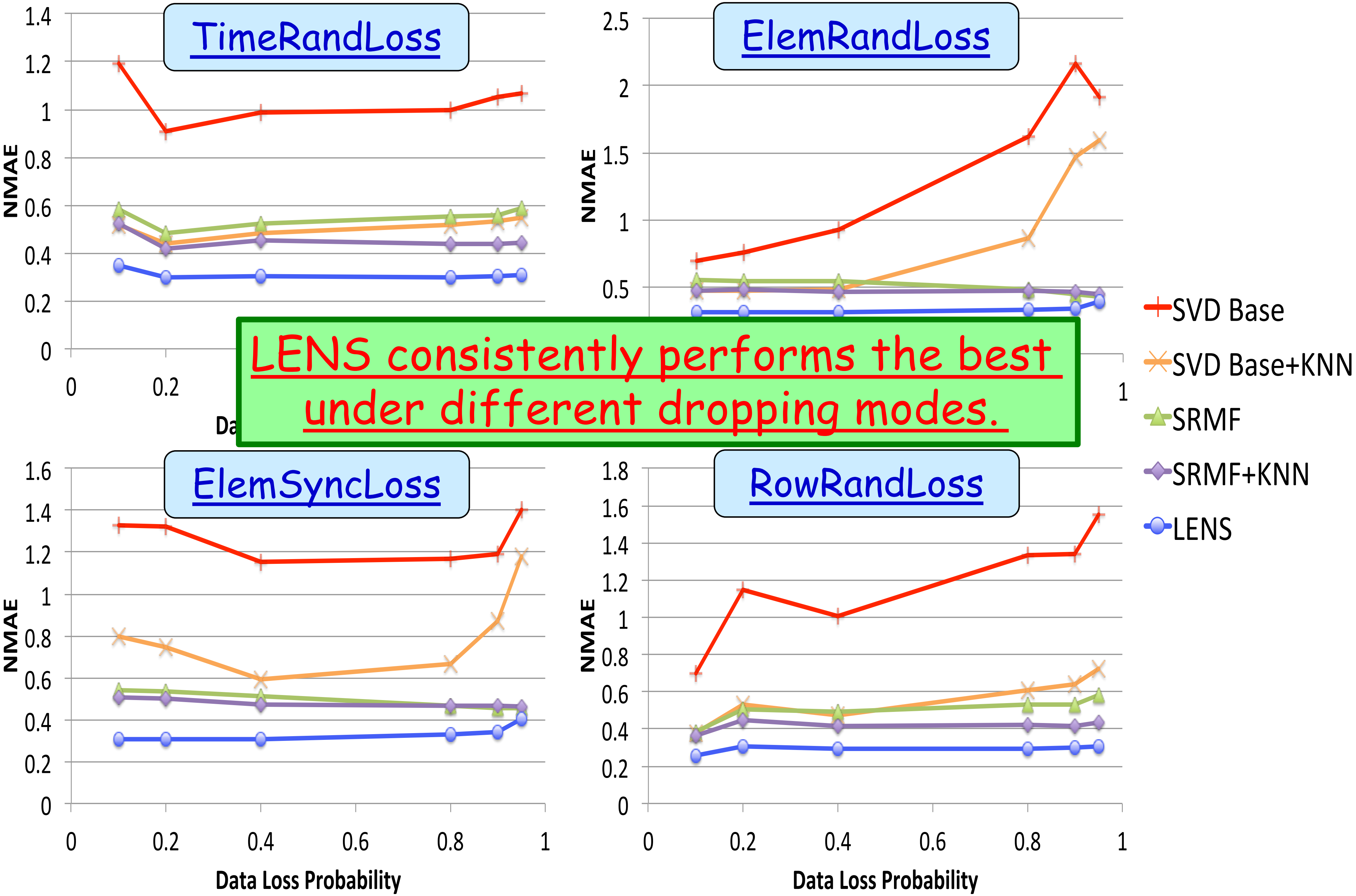
$$\begin{bmatrix} d_{1,1} & d_{1,2} & d_{1,3} & d_{1,4} & & d_{1,n} \\ d_{2,1} & d_{2,2} & & & & \\ d_{3,1} & d_{3,2} & d_{3,3} & d_{3,4} & & d_{3,n} \\ & \vdots & & & \ddots & \\ d_{m,1} & & & & & \end{bmatrix}$$

Dropping Models

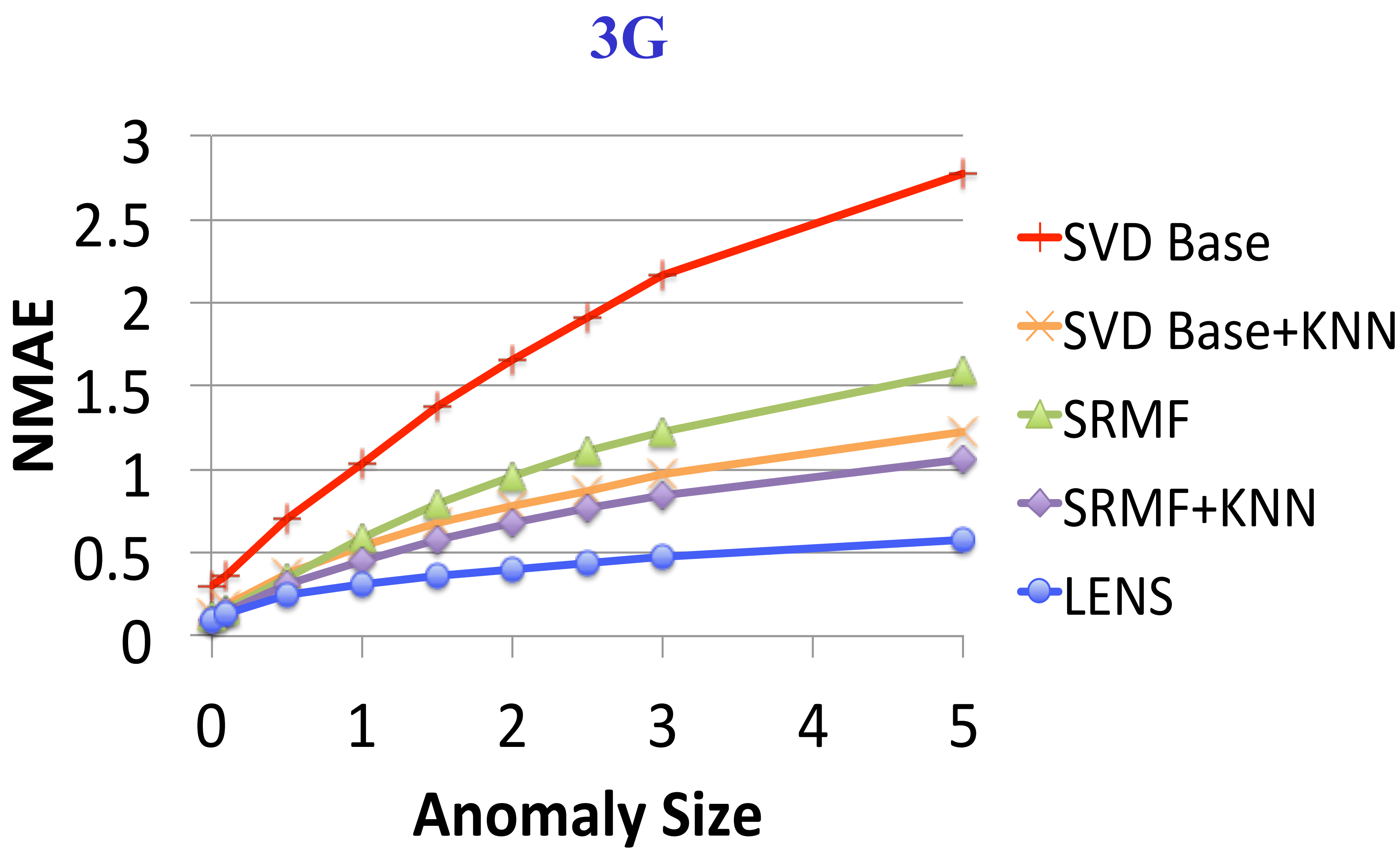
- Element Synchronized Loss
  - Rows are dropped at the same time
  - To emulate certain nodes experience the same lose events at the same time
    - e.g., power outage of an area

$$
\begin{bmatrix}
d_{1,1} & d_{1,2} & d_{1,3} & d_{1,4} & & d_{1,n} \\
 & & & d_{2,4} & \cdots & d_{2,n} \\
 & & & d_{3,4} & & d_{3,n} \\
 & \vdots & & & \ddots & \\
d_{m,1} & d_{m,2} & d_{m,3} & d_{m,4} & & d_{m,n}
\end{bmatrix}
$$

Impact of Dropping models

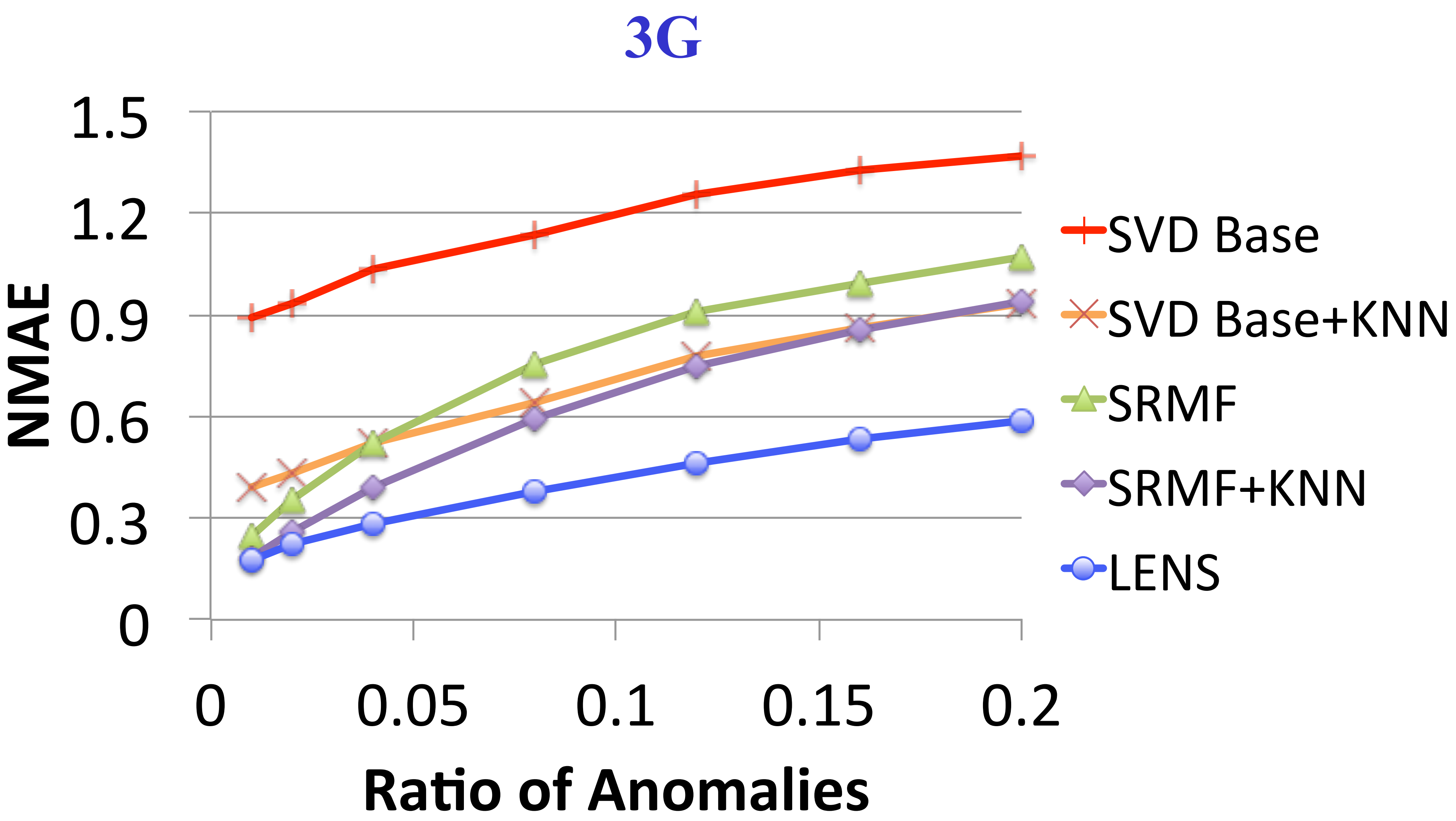


Impact of Anomaly Sizes



The improvement of LENS increases with anomaly sizes.

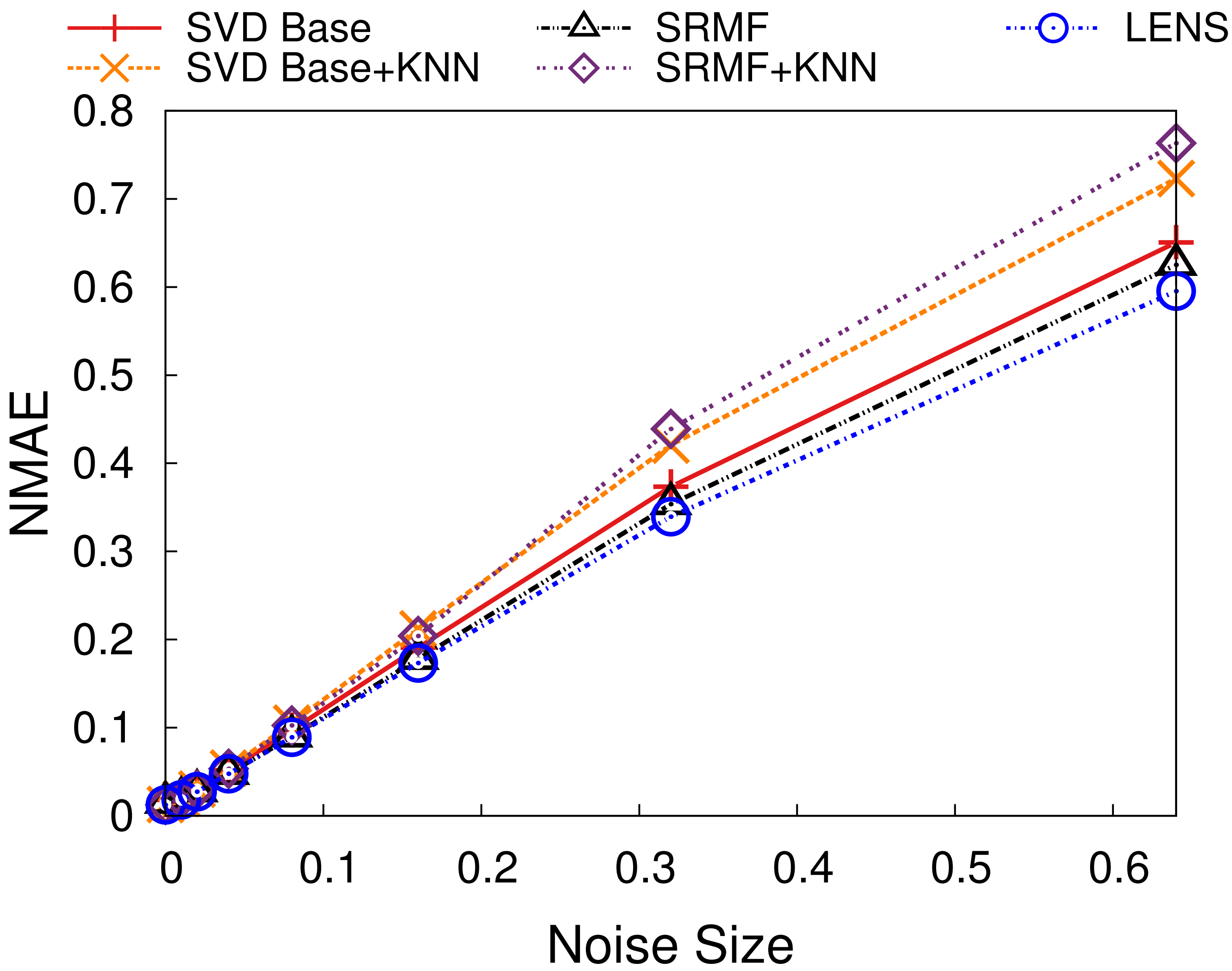
Impact of Number of Anomalies



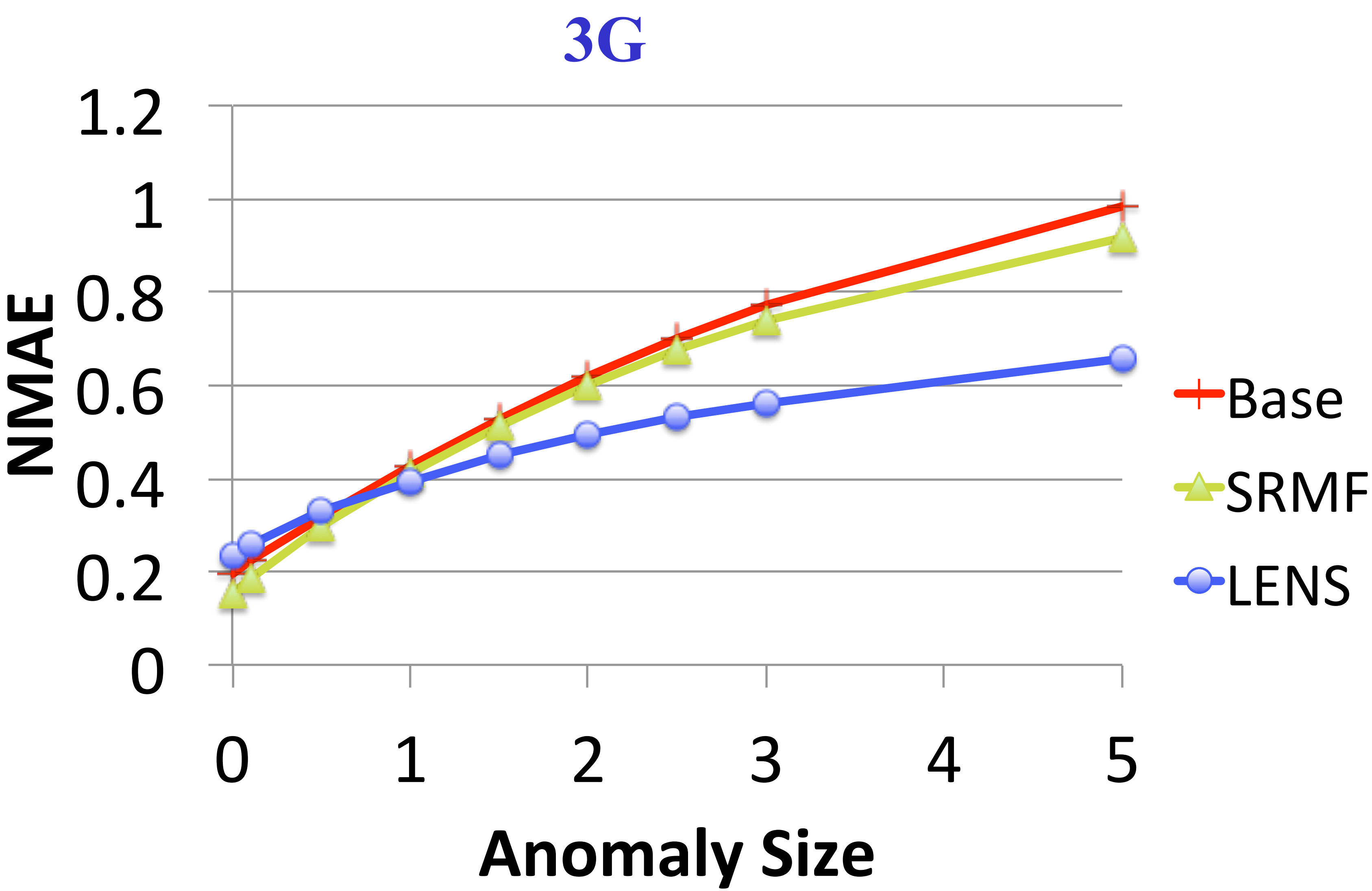
The improvement of LENS increases with # anomalies.

Impact of Noise

CU RSSI

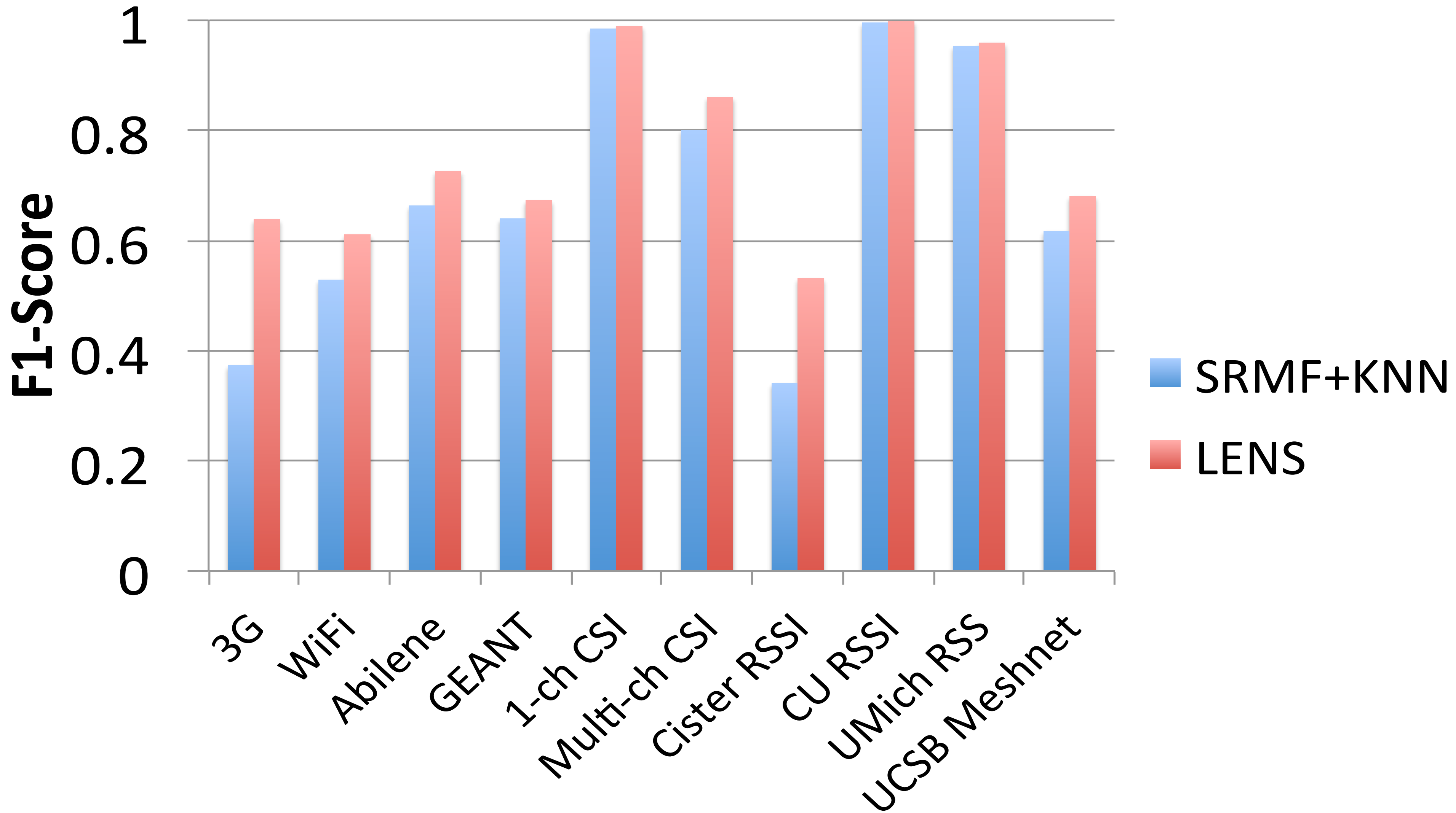


Prediction



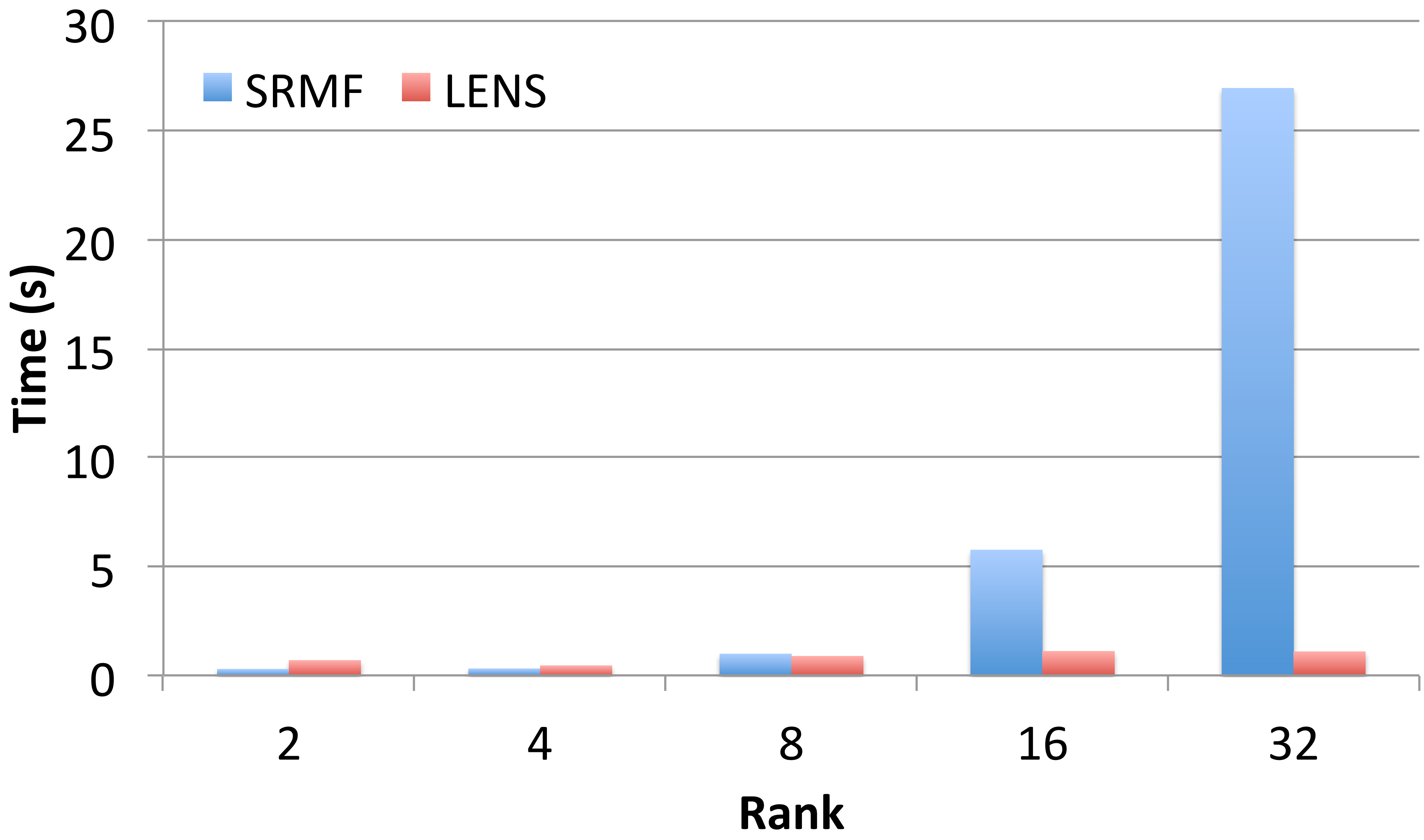
LENS yields the lowest prediction error.

Anomaly Detection



LENS achieves higher anomaly detection accuracy.

Computation Time



Optimization Algorithm

- Alternating Direction Method (ADM)
  - Augmented Lagrangian function

$$
\begin{aligned}
&\mathcal{L}(X, \{X_k\}, Y, Y_0, Z, W, M, \{M_k\}, N, \mu) \\
\triangleq\quad & \alpha\|X\|_* + \beta\|Y\|_1 + \frac{1}{2\sigma}\|Z\|_F^2 \\
& + \frac{\gamma}{2\sigma}\sum_{k=1}^{K}\|P_k X_k Q_k^T - R_k\|_F^2 \\
& + \langle M, D - AX_0 - BY_0 - CZ - W\rangle \\
& + \sum_{k=0}^{K}\langle M_k, X_k - X\rangle \\
& + \langle N, Y_0 - Y\rangle \\
& + \frac{\mu}{2}\cdot\|D - AX_0 - BY_0 - CZ - W\|_F^2 \\
& + \frac{\mu}{2}\cdot\sum_{k=0}^{K}\|X_k - X\|_F^2 \\
& + \frac{\mu}{2}\cdot\|Y_0 - Y\|_F^2
\end{aligned}
$$

Original Objective

Lagrange multiplier

Penalty

$$\langle U, V\rangle \triangleq \sum_{i,j}(U[i,j]\cdot V[i,j])$$

Convergency