Matrix Completion With Cross-Concentrated Sampling: Bridging Uniform Sampling and CUR Sampling

HanQin Cai, Longxiu Huang, Pengyu Li, and Deanna Needell

***Abstract*—While uniform sampling has been widely studied in the matrix completion literature, CUR sampling approximates a low-rank matrix via row and column samples. Unfortunately, both sampling models lack flexibility for various circumstances in real-world applications. In this work, we propose a novel and easy-to-implement sampling strategy, coined Cross-Concentrated Sampling (CCS). By bridging uniform sampling and CUR sampling, CCS provides extra flexibility that can potentially save sampling costs in applications. In addition, we also provide a sufficient condition for CCS-based matrix completion. Moreover, we propose a highly efficient non-convex algorithm, termed Iterative CUR Completion (ICURC), for the proposed CCS model. Numerical experiments verify the empirical advantages of CCS and ICURC against uniform sampling and its baseline algorithms, on both synthetic and real-world datasets.**

***Index Terms*—Cross-concentrated sampling, CUR decomposition, collaborative filtering, image recovery, low-rank matrix, link prediction, matrix completion, recommendation system, sampling strategy.**

I. Introduction

THE problem of *matrix completion* (MC) [1] has received much attention since the last decade. It has arisen in a wide range of applications, e.g., collaborative filtering [2], [3], image processing [4], [5], signal processing [6], [7], genomics [8], [9], multi-task learning [10], system identification [11], and sensor localization [12].

Manuscript received 6 July 2022; revised 28 January 2023; accepted 13 March 2023. Date of publication 23 March 2023; date of current version 30 June 2023. This work was partially supported by AMS Simons Travel under Grants NSF DMS 2011140, NSF DMS 2108479, and NSF DMS 2304489. The authors contributed equally. Recommended for acceptance by L. Li. *(Corresponding author: Longxiu Huang.)*

HanQin Cai is with the Department of Statistics and Data Science and Department of Computer Science, University of Central Florida, Orlando, FL 32816 USA (e-mail: hqcai@ucf.edu).

Longxiu Huang is with the Department of Computational Mathematics, Science and Engineering and Department of Mathematics, Michigan State University, East Lansing, MI 48823 USA (e-mail: huangl3@msu.edu).

Pengyu Li is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: erby1215@g.ucla.edu).

Deanna Needell is with the Department of Mathematics, University of California, Los Angeles, Los Angeles, CA 90095 USA (e-mail: deanna@math.ucla.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2023.3261185, provided by the authors.

Digital Object Identifier 10.1109/TPAMI.2023.3261185

In this article, we study MC under the setting of fixed rank. Consider a rank-r matrix $\boldsymbol{X}$ with observed entries indexed by a set Ω. MC aims to recover the original matrix $\boldsymbol{X}$ from its partial observations. Naturally, we can model this problem as a minimization problem:

$$\begin{aligned} \underset{\widetilde{\boldsymbol{X}}}{\text{minimize}} \quad & \frac{1}{2}\langle \mathcal{P}_\Omega(\boldsymbol{X}-\widetilde{\boldsymbol{X}}), \boldsymbol{X}-\widetilde{\boldsymbol{X}}\rangle \\ \text{subject to} \quad & \operatorname{rank}(\widetilde{\boldsymbol{X}}) = r, \end{aligned} \tag{1}$$

where $\langle , \rangle$ denotes the Frobenius inner product and $\mathcal{P}$ is the sampling operator defined as

$$\mathcal{P}_\Omega(\boldsymbol{X}) = \sum_{(i,j)\in\Omega} [\boldsymbol{X}]_{i,j} \boldsymbol{e}_i \boldsymbol{e}_j^\top. \tag{2}$$

For the success of recovery, the general setting of MC requires the observation set Ω to be sampled via a certain unbiased stochastic process, e.g., uniform sampling with/without replacement [13], [14] or Bernoulli sampling [1] over all matrix entries. While such sample setting has been well studied in both theoretical and empirical aspects [1], [13], [15], [16], [17], [18], [19], it is too restricted in some applications due to hardware, financial, or environmental limitations. For instance, in the application of collaborative filtering, the rows and columns of the data matrix represent the users and rated objects (e.g., movies and merchandise) respectively. The unbiased sample models implicitly assume that all users have the interest to rate all objects with same probability—something that is quite unlikely in practice.

Here we consider *CUR decomposition* as a potential alternative for efficient MC. CUR decomposition is also known as skeleton decomposition [20], [21], which attempts to utilize the self-expressiveness of the data in its low-rank matrix decomposition. There are several different, yet equivalent, formats for CUR decomposition [22]. In particular, we focus on the following CUR format:

$$\boldsymbol{X} = \boldsymbol{C}\boldsymbol{U}^\dagger\boldsymbol{R}, \tag{3}$$

where $\boldsymbol{X}$ is the low-rank data matrix, $\boldsymbol{R}$ and $\boldsymbol{C}$ are selected row and column submatrices of $\boldsymbol{X}$, and $\boldsymbol{U}$ is the intersection submatrix of $\boldsymbol{R}$ and $\boldsymbol{C}$. Obviously, (3) may not hold for arbitrary column and row submatrices. In fact, the following theorem

gives a necessary and sufficient condition for CUR decomposition.

Theorem 1 ([22]). For given row and column submatrices $\boldsymbol{R}$ and $\boldsymbol{C}$, the CUR decomposition (3) holds if and only if $\text{rank}(\boldsymbol{U}) = \text{rank}(\boldsymbol{X}) = r$.

Moreover, with the commonly assumed μ-incoherence (see Assumption 1 in Section II-A later), the following theorem suggests that uniformly selected column and row submatrices are sufficiently good for CUR decomposition.

Theorem 2 ([21], [23], [24]). Let $\boldsymbol{X} \in \mathbb{R}^{n\times n}$ be a rank-r matrix with μ-incoherence.[1] Suppose we sample $|\mathcal{I}| \geq 10\mu_1 r\log(n)$ rows and $|\mathcal{J}| \geq 10\mu_2 r\log(n)$ columns uniformly with replacement. Then $\boldsymbol{U} = [\boldsymbol{X}]_{\mathcal{I},\mathcal{J}}$ satisfies $\text{rank}(\boldsymbol{U}) = \text{rank}(\boldsymbol{X})$ with probability at least $1 - \frac{4r}{n^2}$.

Combining Theorems 1 and 2, one can see that an incoherent low-rank matrix can be recovered from its uniformly sampled rows and columns via CUR decomposition. In this sense, CUR decomposition can be viewed as an MC solver [9], [25]. We call the corresponding sampling model *CUR sampling*, i.e., full observation on the sampled rows and columns. Nevertheless, the model of CUR sampling is also too restricted in many applications, especially with larger-scale problems. For instance, in the application of larege-scale collaborative filtering, CUR sampling implicitly assumes that some users rate *all* objects and some objects are rated by *all* users, which is clearly impractical.

In the era of Big Data, it is urgent to explore some efficient sampling models that suit various real-world circumstances. While both the uniform sampling and CUR sampling have limits in applications, the blank space between them leads to a more flexible and attainable sampling strategy. In this work, we propose a novel sampling model, coined *Cross-Concentrated Sampling* (CCS), to bridge the aforementioned uniform sampling and CUR sampling. Our approach allows partial observations on selected row and column submatrices, making it much practical in many applications.

A. Related Work and Contributions

1) Matrix Completion: The pioneering work [1] studies the matrix completion problem with the Bernoulli sampling model. By relaxing the non-convex problem to a convex nuclear norm minimization, it shows that sampling $\mathcal{O}(rn^{1.2}\log(n))$ entries is sufficient for exact recovery with high probability, which is subsequently improved to $\mathcal{O}(rn\log^2(n))$ in [26]. Another study [13] focuses on the uniform sampling model, and achieves the same improved sample complexity $\mathcal{O}(rn\log^2(n))$. The standard algorithms for solving the nuclear norm minimization are semidefinite programming [27] and singular value thresholding [15]. More recently, many non-convex algorithms that aim at the original non-convex problem have also been studied: [28], [29] use the technique of singular value projection (SVP) and provide strong empirical performance; however, the theoretical sample complexity is high as $\mathcal{O}(r^5 n\log^3(n))$. The works [18], [30] are based on alternating minimization and have the sample complexities $\mathcal{O}(r^{2.5}n\log(n)\log(\frac{1}{\varepsilon}))$ and $\mathcal{O}(rn\log(n)\log(\frac{1}{\varepsilon}))$ respectively, where ε is the desired accuracy. Note that the term $\log(\frac{1}{\varepsilon})$ is introduced since [18], [30] require iterative resampling. A series of work [17], [19], [31], [32] studies the (modified) gradient descent methods on Grassmannian manifold where the sharpest sample complexity is $\mathcal{O}(r^2 n\log(n))$. [33], [34] focus on fast Riemannian optimization approaches and achieve the sample complexity $\mathcal{O}(r^2 n\log^2(n))$. Nevertheless, all these algorithms are designed for Bernoulli or uniform sampling models.

Note that [16] shows the equivalence between the Bernoulli and uniform sampling models in matrix completion. Thus, for the ease of presentation, we only discuss the uniform sampling model in this paper; however, we emphasize that our approach can be easily extended to bridge between Bernoulli sampling and CUR sampling for a similar result.

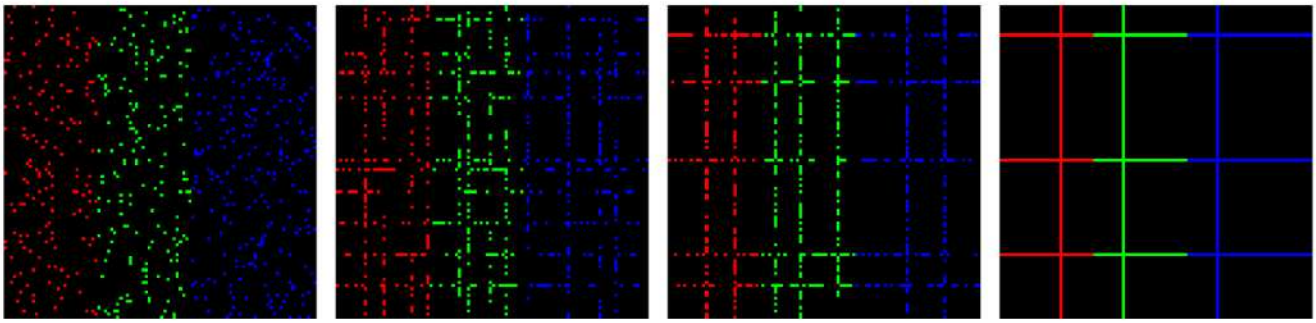
2) CUR Decomposition: For a given rank-r matrix $\boldsymbol{X} \in \mathbb{R}^{n\times n}$, CUR decomposition represents $\boldsymbol{X}$ by its submatrices. There are two different versions of CUR decomposition. Set $\boldsymbol{C} = [\boldsymbol{X}]_{:,\mathcal{J}}$ and $\boldsymbol{R} = [\boldsymbol{X}]_{\mathcal{I},:}$. One type of CUR decomposition is of the form (3). Another version expresses $\boldsymbol{X}$ as $\boldsymbol{C}\boldsymbol{C}^{\dagger}\boldsymbol{X}\boldsymbol{R}^{\dagger}\boldsymbol{R}$. The equivalence of these two distinct CUR decompositions is proved in [22]. Ensurance of the exact CUR decomposition is equivalent to the condition that $\text{rank}(\boldsymbol{U}) = \text{rank}(\boldsymbol{X})$ with $\boldsymbol{U} = [\boldsymbol{X}]_{\mathcal{I},\mathcal{J}}$. There are deterministic [35], [36], [37] and random [21], [38], [39], [40], [41], [42], [43] methods to select the row and column subsets to form CUR decomposition. Deterministic sampling needs to sample fewer rows and columns, e.g., $\mathcal{O}(r)$ to guarantee CUR decomposition but it needs to access the full data and is more computationally costly. Random sampling is usually computationally cheaper, but it requires more rows and columns. In the literature, there are three popularly used random sampling distributions: uniform [21], column/row length [38], and leverage scores [39]. Compared with the other two, uniform sampling is the easiest and cheapest to implement and does not need to access the full data, but it may fail to provide good results for a generic matrix [44]. However, as discussed, if the given matrix is incoherent, then uniform sampling can guarantee good performance [21], [23], [24], [45].

Although the application of CUR decomposition on MC has already been discussed in [9], [25], both papers require full observation on the selected row and column submatrices, which, as discussed, is too restricted in some applications.

3) Contributions: This paper bridges the uniform sampling and CUR sampling for matrix completion (MC) problems. Under the commonly used incoherence assumption, we propose a novel sampling model, coined Cross-Concentrated Sampling (CCS). To summarize, our main contributions are as follows:

1) We propose a flexible and attainable sampling model, coined CCS, for MC that bridges uniform sampling and CUR sampling (see Procedure 1).
2) We establish a sufficient condition for exact data recovery from the proposed CCS model, specifically $\mathcal{O}(r^2 n\log^2(n))$ samples are sufficient to exactly reconstruct the missing data with a high probability (see Theorem 4).

[1] To simplify our expression, we assume the matrices are square throughout the paper but all the results can be generalized to rectangular matrices.



(a) Uniform Sampling (b) CCS–Less Concentrated (c) CCS–More Concentrated (d) CUR Sampling

Fig. 1. Visual illustrations of different sampling schemes. From left to right, sampling methods change from the uniform sampling style to the CUR sampling style with the same total observations rate. Colored pixels indicate observed entries, while black pixels mean missing entries.

TABLE I
TABLE OF NOTATION

NOT.	DESCRIPTION
$\mathcal{P}_\Omega$	sampling operator on the set Ω (see (2))
$\mathcal{I}$	row indices for the row submatrix $\boldsymbol{R}$
$\mathcal{J}$	column indices for the column submatrix $\boldsymbol{C}$
$\Omega_{\boldsymbol{R}}$	indices set of the samples on the submatrix $\boldsymbol{R}$
$\Omega_{\boldsymbol{C}}$	indices set of the samples on the submatrix $\boldsymbol{C}$
δ	percentage of sampled columns or rows
p	uniform observation rate on the submatrices
s	overall observation size on the full matrix
α	overall observation rate on the full matrix

3) We design a highly efficient non-convex algorithm, dubbed Iterative CUR Completion (ICURC), for solving CCS-based MC problem (see Algorithm 2). In particular, ICURC costs merely $\mathcal{O}(nr(|\mathcal{I}|+|\mathcal{J}|))$ flops provided $|\mathcal{I}|, |\mathcal{J}| \ll n$.
4) We demonstrate the effectiveness and efficiency of the CCS model and the corresponding algorithm on both synthetic and real-world datasets (see Section IV).

B. Notation

Given matrices $\boldsymbol{X} \in \mathbb{R}^{n\times n}$, $[\boldsymbol{X}]_{i,j}$, $[\boldsymbol{X}]_{\mathcal{I},:}$, $[\boldsymbol{X}]_{:,\mathcal{J}}$, and $[\boldsymbol{X}]_{\mathcal{I},\mathcal{J}}$ denote the (i,j)-th entry of $\boldsymbol{X}$, the row submatrix with row indices $\mathcal{I}$, the column submatrix with column indices $\mathcal{J}$, and the submatrix of $\boldsymbol{X}$ with row indices $\mathcal{I}$ and column indices $\mathcal{J}$, respectively. $\|\boldsymbol{X}\|_{\mathrm{F}} := (\sum_{i,j}[\boldsymbol{X}]_{i,j}^2)^{1/2}$ denotes the Frobenius norm of $\boldsymbol{X}$, $\|\boldsymbol{X}\|_{2,\infty} := \max_i(\sum_j [\boldsymbol{X}]_{i,j}^2)^{1/2}$ denotes the largest row-wise ℓ_2-norm, $\|\boldsymbol{X}\|_\infty = \max_{i,j}|[\boldsymbol{X}]_{i,j}|$, $\boldsymbol{X}^\dagger$ represents the Moore–Penrose inverse of $\boldsymbol{X}$, and $\boldsymbol{X}^\top$ is the transpose of $\boldsymbol{X}$. For $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^{n\times n}$, $\langle \boldsymbol{X}, \boldsymbol{Y}\rangle = \sum_{i,j}[\boldsymbol{X}]_{i,j}[\boldsymbol{Y}]_{i,j}$ denotes the Frobenius inner product of $\boldsymbol{X}$ and $\boldsymbol{Y}$. The symbol $[n]$ denotes the set $\{1,\ldots,n\}$ for all $n \in \mathbb{Z}^+$. $\mathcal{I}\times[n]$ denotes the set $\{(i,j): i\in\mathcal{I}, j\in[n]\}$ and $[n]\times\mathcal{J}$ denotes the set $\{(i,j): i\in[n], j\in\mathcal{J}\}$. Unless otherwise specified, the term *uniform sampling* refers to uniform sampling with replacement throughout the paper. Additionally, some important symbols are summarized in Table I.

II. PROPOSED MODEL

We aim to design an efficient and effective sampling strategy for varying circumstances in real-world applications. Motivated by the example of collaborative filtering and CUR decomposition, we propose a novel sampling model that samples the entries concentrated on a subset of rows and columns of the original data matrix. Formally, let $\boldsymbol{R} = [\boldsymbol{X}]_{\mathcal{I},:}$ and $\boldsymbol{C} = [\boldsymbol{X}]_{:,\mathcal{J}}$ be selected (by indices sets $\mathcal{I}$ and $\mathcal{J}$) row and column submatrices of the data matrix $\boldsymbol{X}$, respectively. Next, we uniformly (with replacement) sample entries on $\boldsymbol{R}$ and $\boldsymbol{C}$, i.e., the samples are concentrated on the selected row and column submatrices. Since the visualization (Fig. 1) of the submatrices $\boldsymbol{R}$ and $\boldsymbol{C}$ together looks like crosses, we name this sampling model *Cross-Concentrated Sampling* (CCS). Moreover, we illustrate CCS against uniform sampling and CUR sampling in Fig. 1. One can see that CCS becomes CUR sampling if samples are dense enough to fully observe the submatrices, and CCS becomes uniform sampling if all rows and columns are selected into the submatrices.

We denote the indices sets of the cross-concentrated samples by $\Omega_{\boldsymbol{R}}$ and $\Omega_{\boldsymbol{C}}$ with respect to the notation of the submatrices. The samples in the intersection submatrix $\boldsymbol{U}$ are directly inherited from $\Omega_{\boldsymbol{R}} \cup \Omega_{\boldsymbol{C}}$:[2]

$$\Omega_{\boldsymbol{U}} = \{(i,j) \in \Omega_{\boldsymbol{R}} \cup \Omega_{\boldsymbol{C}} \mid i \in \mathcal{I} \text{ and } j \in \mathcal{J}\}. \tag{4}$$

Thus, the expected observation rate of $\Omega_{\boldsymbol{U}}$ is the sum of $\Omega_{\boldsymbol{R}}$'s and $\Omega_{\boldsymbol{C}}$'s observation rates. Our task is recovering the underlying rank-r $\boldsymbol{X}$ from the observations on $\Omega_{\boldsymbol{R}} \cup \Omega_{\boldsymbol{C}}$:

$$\begin{aligned} \underset{\widetilde{\boldsymbol{X}}}{\text{minimize}} \quad & \frac{1}{2}\langle \mathcal{P}_{\Omega_{\boldsymbol{R}}\cup\Omega_{\boldsymbol{C}}}(\boldsymbol{X}-\widetilde{\boldsymbol{X}}), \boldsymbol{X}-\widetilde{\boldsymbol{X}}\rangle \\ \text{subject to} \quad & \operatorname{rank}(\widetilde{\boldsymbol{X}}) = r, \end{aligned} \tag{5}$$

where $\mathcal{P}$ is the sampling operator defined in (2). Note that $\mathcal{P}$ is not a projection operator if there are repeated samples in the

[2]We are abusing set notation here. Since we use "sampling with replacement" and thus allow repeated samples, $\Omega_{\boldsymbol{R}}$ and $\Omega_{\boldsymbol{C}}$ are not precisely sets, and $\Omega_{\boldsymbol{R}} \cup \Omega_{\boldsymbol{C}}$ is actually $\Omega_{\boldsymbol{R}} + \Omega_{\boldsymbol{C}}$.

Procedure 1: Cross-Concentrated Sampling (CCS).

1: **Input:** $\boldsymbol{X}$: access to underlying low-rank matrix.
2: Uniformly choose row and column indices $\mathcal{I}, \mathcal{J}$.
3: Set $\boldsymbol{R} = [\boldsymbol{X}]_{\mathcal{I},:}$ and $\boldsymbol{C} = [\boldsymbol{X}]_{:,\mathcal{J}}$.
4: Uniformly sample entries in $\boldsymbol{R}$ and $\boldsymbol{C}$, then record the sampled locations as $\Omega_{\boldsymbol{R}}$ and $\Omega_{\boldsymbol{C}}$, respectively.
5: **Output:** $[\boldsymbol{X}]_{\Omega_{\boldsymbol{R}} \cup \Omega_{\boldsymbol{C}}}, \Omega_{\boldsymbol{R}}, \Omega_{\boldsymbol{C}}, \mathcal{I}, \mathcal{J}$.

indices set. Hence, $\langle \mathcal{P}_\Omega(\boldsymbol{X}), \boldsymbol{X} \rangle$ may not equal to $\|\mathcal{P}_\Omega(\boldsymbol{X})\|_{\mathrm{F}}^2$ in our formula.

Remark 1. For successful recovery, the rows and columns we sample from must span the full row and column space of $\boldsymbol{X}$. In other words, we require $\operatorname{rank}(\boldsymbol{U}) = \operatorname{rank}(\boldsymbol{X})$ where $\boldsymbol{U} = [\boldsymbol{X}]_{\mathcal{I},\mathcal{J}}$ is the intersection submatrix of $\boldsymbol{R}$ and $\boldsymbol{C}$. By Theorem 2, this condition can be achieved by uniformly sampling $\mathcal{O}(r \log(n))$ row and column indices if $\boldsymbol{X}$ is incoherent. One may select as low as $\mathcal{O}(r)$ rows and columns based on prior knowledge in some applications. We consider this as a necessary condition for CCS-based matrix completion.

Example 1. Consider the famous Netflix problem where each row of the data matrix represents a user and each column represents a movie [2]. The CCS model randomly selects some users and has them randomly rate a sufficient amount of movies (perhaps through monetary incentive), then randomly selects some movies and has adequate number of users rate them (perhaps through website promotion). If CCS has access to some background information of the users, then we can select fewer but representative users from various backgrounds for the survey. Similarly, fewer but representative movies of each category will be promoted for enough ratings. Compare to uniform sampling, CCS is able to collect desired data points much more efficiently. Compared to CUR sampling, CCS is more realistic, as CUR sampling asks all the selected users to rate all existing movies, and all existing users rate all the selected movies.

In summary, we present the CCS model as Procedure 1 for low-rank matrices with incoherence.

A. Theoretical Results

In this section, we study the CCS model from a theoretical perspective. In particular, we provide a sufficient condition to guarantee the uniqueness of solutions with the samples generated by Procedure 1. The proofs are deferred to Section V.

We start with the formal expression of the widely used incoherence assumption.

Assumption 1 ($\{\mu_1, \mu_2\}$-incoherence). Let $\boldsymbol{X} \in \mathbb{R}^{n \times n}$ be a rank-r matrix. $\boldsymbol{X}$ is $\{\mu_1, \mu_2\}$-incoherent if

$$\|\boldsymbol{W}\|_{2,\infty} \leq \sqrt{\frac{\mu_1 r}{n}} \quad \text{and} \quad \|\boldsymbol{V}\|_{2,\infty} \leq \sqrt{\frac{\mu_2 r}{n}}$$

for some constants μ_1 and μ_2, where $\boldsymbol{W}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ is the compact singular value decomposition (SVD) of $\boldsymbol{X}$. In some context, we use the term μ-incoherence where $\mu := \max\{\mu_1, \mu_2\}$ for simplicity.

Next, we present a variant of [24, Theorem 3.5] that shows how the matrix properties are transformed to its uniformly sampled row submatrix, which is a keystone to our main theorem. Similar results hold for a uniformly sampled column submatrix as well.

Lemma 3. Suppose that $\boldsymbol{X} \in \mathbb{R}^{n \times n}$ is a rank-r matrix that satisfies Assumption 1. Let κ denote the condition number of $\boldsymbol{X}$. Suppose that the indices set $\mathcal{I} \subseteq [n]$ is chosen by sampling uniformly without replacement to yield $\boldsymbol{R} = [\boldsymbol{X}]_{\mathcal{I},:}$. If $|\mathcal{I}| \geq \mu_1 r^2 \log^2(n)$, then the following conditions hold with probability at least $1 - \frac{r}{n^{0.4r\log(n)}}$:

$$\mu_{1\boldsymbol{R}} \leq 4\kappa^2 \mu_1, \quad \mu_{2\boldsymbol{R}} \leq \mu_2, \quad \kappa_{\boldsymbol{R}} \leq 2\sqrt{\mu_1 r}\kappa,$$

where $\{\mu_{1\boldsymbol{R}}, \mu_{2\boldsymbol{R}}\}$ and $\kappa_{\boldsymbol{R}}$ are the incoherence parameters and the condition number of $\boldsymbol{R}$ respectively.

Now, we are ready to present our main theorem and its proof is deferred to Section V-B.

Theorem 4. Suppose $\boldsymbol{X} \in \mathbb{R}^{n \times n}$ is rank-r matrix that satisfies Assumption 1. Let κ denote the condition number of $\boldsymbol{X}$. Suppose that $\mathcal{I}, \mathcal{J} \subseteq [n]$ are chosen uniformly with replacement to yield $\boldsymbol{R} = [\boldsymbol{X}]_{\mathcal{I},:}$ and $\boldsymbol{C} = [\boldsymbol{X}]_{:,\mathcal{J}}$. Suppose $\Omega_{\boldsymbol{R}}$ and $\Omega_{\boldsymbol{C}}$ are sampled uniformly with replacement. If

$$\begin{aligned}
|\mathcal{I}| &\geq 512\beta\kappa^2 r^2 \mu_1 \mu_2 \log^2(n), \\
|\mathcal{J}| &\geq 512\beta\kappa^2 r^2 \mu_1 \mu_2 \log^2(n), \\
|\Omega_{\boldsymbol{R}}| &\geq 128\beta\kappa^2 r^2 \mu_1 \mu_2 (n + |\mathcal{I}|) \log^2(2n), \\
|\Omega_{\boldsymbol{C}}| &\geq 128\beta\kappa^2 r^2 \mu_1 \mu_2 (n + |\mathcal{J}|) \log^2(2n),
\end{aligned}$$

for some absolute constant $\beta > 1$, then $\boldsymbol{X}$ can be uniquely determined from $\Omega_{\boldsymbol{R}} \cup \Omega_{\boldsymbol{C}}$ with probability at least

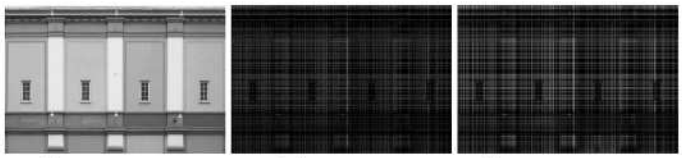
$$1 - \frac{2r}{n^{0.4r\log(n)}} - \frac{2}{n^{2\beta^{0.5}-2}} - \sum_{i=1}^{2} \frac{6\log(n)}{(n + \mu_i r^2 \log^2(n))^{2\beta-2}}.$$

Theorem 4 shows a sufficient sample complexity for CCS-based MC is $\mathcal{O}(r^2 n \log^2(n))$. Compared to the state-of-the-art uniform-sampling-based MC approach [32] that requires $\mathcal{O}(r^2 n \log(n))$ samples, our result is merely a factor of $\log(n)$ worse under the same incoherence assumption[3]. Moreover, the same condition also guarantees the uniqueness of the solution. Essentially, Theorem 4 states that concentrating the samples into some properly chosen rows and columns will not blow up the required sampling complexity. Hence, based on the circumstance, one can choose how concentrated the samples are, which can potentially simplify the sampling process in some applications, e.g., as we have discussed in Example 1.

III. A Non-Convex Solver

In this section, we discuss how to effectively and efficiently solve the CCS-based MC problem. First, we consider directly applying the existing uniform-sampling-based MC algorithms.

[3]Some works achieve better sample complexities by utilizing additional assumption(s) that we do not use. For example, [13], [26] obtain a sample complexity $\mathcal{O}(rn\log^2(n))$, but also use the additional assumption: $\|\boldsymbol{W}\boldsymbol{V}^\top\|_\infty \leq \mu_3\sqrt{r}/n$ for some constant μ_3.



(a) Groundtruth (b) Observed (c) Reconstruction

Fig. 2. Visual results for image inpainting from the CCS-based samples via ScalePGD algorithm [32]. See a more detailed setting in Section IV-B.

Unfortunately, it turns out that the uniform-sampling-based algorithms are not suitable for the CCS model. For example, as shown in Fig. 2, we apply the state-of-the-art ScaledPGD [32] on a CCS-based image recovery problem and the result is not visually desirable. Therefore, we must develop new algorithm(s) for the proposed CCS model.

Recall that CCS samples uniformly on the selected row and column submatrices. Thus, a natural and simple approach is solving the MC problem in two steps:

1) Applying certain off-the-shelf MC algorithms to recover the submatrices $\boldsymbol{R}$ and $\boldsymbol{C}$ separately. Note that $\Omega_{\boldsymbol{R}}$ and $\Omega_{\boldsymbol{C}}$ are uniformly sampled in $\boldsymbol{R}$ and $\boldsymbol{C}$. Thus, any uniform-sampling-based MC solver will work, provided enough samples.
2) Applying the standard CUR decomposition, i.e., (3), to recover $\boldsymbol{X}$ from the fully reconstructed $\boldsymbol{R}$ and $\boldsymbol{C}$.

This approach is named Two-Step Completion (TSC) and is later summarized as Algorithm 3 in Section V. However, this algorithm is not utilizing the samples in $\boldsymbol{C}$ when recovering $\boldsymbol{R}$ and vice versa. Thus, one does not expect TSC to be a highly effective solver for (5).

To take full advantage of the CCS structure, we propose a novel non-convex algorithm, coined Iterative CUR Completion (ICURC), for the CCS-based MC problem. ICURC is built upon the framework of projected gradient descent [28]. Notice that

$$\begin{aligned}&\mathbb{E}(\mathcal{P}_{\Omega_{\boldsymbol{R}}\cup\Omega_{\boldsymbol{C}}}(\boldsymbol{X}))\\&=\mathbb{E}(\mathcal{P}_{\Omega_{\boldsymbol{R}}}(\boldsymbol{X}))+\mathbb{E}(\mathcal{P}_{\Omega_{\boldsymbol{C}}}(\boldsymbol{X}))\\&=p_1\mathcal{P}_{\mathcal{I}\times[n]}(\boldsymbol{X})+p_2\mathcal{P}_{[n]\times\mathcal{J}}(\boldsymbol{X})\\&=p_1\mathcal{P}_{\mathcal{I}\times\mathcal{J}^c}(\boldsymbol{X})+p_2\mathcal{P}_{\mathcal{I}^c\times\mathcal{J}}(\boldsymbol{X})+(p_1+p_2)\mathcal{P}_{\mathcal{I}\times\mathcal{J}}(\boldsymbol{X}),\end{aligned}\tag{6}$$

where $p_1=\frac{|\Omega_{\boldsymbol{R}}|}{n|\mathcal{I}|}$, $p_2=\frac{|\Omega_{\boldsymbol{C}}|}{n|\mathcal{J}|}$, $\mathcal{J}^c=[n]\setminus\mathcal{J}:=\{j\in[n]:j\notin\mathcal{J}\}$, and $\mathcal{I}^c=[n]\setminus\mathcal{I}:=\{i\in[n]:i\notin\mathcal{I}\}$. Therefore, in each iteration, we divide the updating of $\boldsymbol{R}$ and $\boldsymbol{C}$ into three part: $[\boldsymbol{R}]_{:,\mathcal{J}^c}$, $[\boldsymbol{C}]_{\mathcal{I}^c,:}$, and $\boldsymbol{U}$. Additionally, to enforce the rank constrain on $\boldsymbol{X}$, we project the updated $\boldsymbol{U}$ to its best rank-r approximation. Specifically, we perform a step of gradient descent on $[\boldsymbol{R}]_{:,\mathcal{J}^c}$, $[\boldsymbol{C}]_{\mathcal{I}^c,:}$, and $\boldsymbol{U}$ via the following formula:

$$\begin{aligned}[\boldsymbol{R}_{k+1}]_{:,\mathcal{J}^c}&:=[\boldsymbol{X}_k]_{\mathcal{I},\mathcal{J}^c}+\eta_R[\mathcal{P}_{\Omega_{\boldsymbol{R}}}(\boldsymbol{X}-\boldsymbol{X}_k)]_{\mathcal{I},\mathcal{J}^c},\\[\boldsymbol{C}_{k+1}]_{\mathcal{I}^c,:}&:=[\boldsymbol{X}_k]_{\mathcal{I}^c,\mathcal{J}}+\eta_C[\mathcal{P}_{\Omega_{\boldsymbol{C}}}(\boldsymbol{X}-\boldsymbol{X}_k)]_{\mathcal{I}^c,\mathcal{J}},\end{aligned}\tag{7}$$

and

$$\boldsymbol{U}_{k+1}:=\mathcal{H}_r\left([\boldsymbol{X}_k]_{\mathcal{I},\mathcal{J}}+\eta_U[\mathcal{P}_{\Omega_{\boldsymbol{R}}\cup\Omega_{\boldsymbol{C}}}(\boldsymbol{X}-\boldsymbol{X}_k)]_{\mathcal{I},\mathcal{J}}\right),\tag{8}$$

Algorithm 2: Iterative CUR Completion (ICURC) for CCS.

Input: $[\boldsymbol{X}]_{\Omega_{\boldsymbol{R}}\cup\Omega_{\boldsymbol{C}}}$: observed data; $\Omega_{\boldsymbol{R}},\Omega_{\boldsymbol{C}}$: observation locations; $\mathcal{I},\mathcal{J}$: row and column indices that define $\boldsymbol{R}$ and $\boldsymbol{C}$ respectively; η_R,η_C,η_U: step sizes; r: target rank; ε: target precision level.
$\boldsymbol{X}_0=\boldsymbol{0}$
$\mathcal{I}^c=[n]\setminus\mathcal{I}$, $\mathcal{J}^c=[n]\setminus\mathcal{J}$, $k=0$
while $e_k>\varepsilon$ **do** ▷e_k is defined in (10)
  $[\boldsymbol{R}_{k+1}]_{:,\mathcal{J}^c}=[\boldsymbol{X}_k]_{\mathcal{I},\mathcal{J}^c}+\eta_R[\mathcal{P}_{\Omega_{\boldsymbol{R}}}(\boldsymbol{X}-\boldsymbol{X}_k)]_{\mathcal{I},\mathcal{J}^c}$
  $[\boldsymbol{C}_{k+1}]_{\mathcal{I}^c,:}=[\boldsymbol{X}_k]_{\mathcal{I}^c,\mathcal{J}}+\eta_C[\mathcal{P}_{\Omega_{\boldsymbol{C}}}(\boldsymbol{X}-\boldsymbol{X}_k)]_{\mathcal{I}^c,\mathcal{J}}$
  $\boldsymbol{U}_{k+1}=\mathcal{H}_r([\boldsymbol{X}_k]_{\mathcal{I},\mathcal{J}}+\eta_U[\mathcal{P}_{\Omega_{\boldsymbol{R}}\cup\Omega_{\boldsymbol{C}}}(\boldsymbol{X}-\boldsymbol{X}_k)]_{\mathcal{I},\mathcal{J}})$
  $[\boldsymbol{R}_{k+1}]_{:,\mathcal{J}}=\boldsymbol{U}_{k+1}$
  $[\boldsymbol{C}_{k+1}]_{\mathcal{I},:}=\boldsymbol{U}_{k+1}$
  $\boldsymbol{X}_{k+1}=\boldsymbol{C}_{k+1}\boldsymbol{U}_{k+1}^{\dagger}\boldsymbol{R}_{k+1}$ ▷Do not compute, see (11)
  $k=k+1$
end while
Output: $\boldsymbol{C}_k,\boldsymbol{U}_k,\boldsymbol{R}_k$: CUR components of $\boldsymbol{X}$.

where $\eta_R,\eta_C,\eta_U>0$ are the step sizes, and $\mathcal{H}_r$ is the rank-r truncated SVD operator. We then set $[\boldsymbol{R}_{k+1}]_{:,\mathcal{J}}=[\boldsymbol{C}_{k+1}]_{\mathcal{I},:}=\boldsymbol{U}_{k+1}$. Hence, updated via CUR decomposition, the approximated data matrix

$$\boldsymbol{X}_{k+1}=\boldsymbol{C}_{k+1}\boldsymbol{U}_{k+1}^{\dagger}\boldsymbol{R}_{k+1}\tag{9}$$

is also rank-r. Although CUR decomposition is not the most accurate low-rank approximation, it is close enough for our iterative algorithm.

With the initial guess $\boldsymbol{X}_0=0$, ICURC runs the above steps iteratively until convergence. In particular, we set the stopping criterion to be $e_k\le\varepsilon$ where

$$e_k=\frac{\langle\mathcal{P}_{\Omega_{\boldsymbol{R}}\cup\Omega_{\boldsymbol{C}}}(\boldsymbol{X}-\boldsymbol{X}_k),\boldsymbol{X}-\boldsymbol{X}_k\rangle}{\langle\mathcal{P}_{\Omega_{\boldsymbol{R}}\cup\Omega_{\boldsymbol{C}}}(\boldsymbol{X}),\boldsymbol{X}\rangle}\tag{10}$$

and ε is the targeted accuracy. We summarize the proposed non-convex algorithm as Algorithm 2.

Remark 2. Inspired by [13, Theorem 7], we recommend the step size $\eta_R=\frac{1}{p_1}$, $\eta_C=\frac{1}{p_2}$, and $\eta_U=\frac{1}{p_1+p_2}$ for Algorithm 2, where p_1 and p_2 are the observation rates of $\Omega_{\boldsymbol{R}}$ and $\Omega_{\boldsymbol{C}}$, respectively.

A. Efficient Implementation

We provide the implementation details and the breakdown of the computational costs for Algorithm 2. The steps of updating $[\boldsymbol{R}_{k+1}]_{\mathcal{I},\mathcal{J}^c}$ and $[\boldsymbol{C}_{k+1}]_{\mathcal{I}^c,\mathcal{J}}$, i.e., (7), cost $\mathcal{O}(|\Omega_{\boldsymbol{R}}|+|\Omega_{\boldsymbol{C}}|-|\Omega_{\boldsymbol{U}}|)$ flops as we only update the observed locations. Note that the intersection matrix $\boldsymbol{U}$ is $|\mathcal{I}|\times|\mathcal{J}|$. Thus, in (8) and (9), computing the truncated SVD and pseudo-inverse of $\boldsymbol{U}_{k+1}$ costs $\mathcal{O}(r|\mathcal{I}||\mathcal{J}|)$ flops. Calculating $\boldsymbol{X}_{k+1}$ in (9) seems expensive at first sight; fortunately, we do not actually have to form the whole $n\times n$ matrix. Looking back at (7) and (8), we only use the selected rows and columns from $\boldsymbol{X}_k$ and they can be efficiently

obtained via

$$[\boldsymbol{X}_k]_{\mathcal{I},\mathcal{J}^c} = [\boldsymbol{C}_k]_{\mathcal{I},:}\boldsymbol{U}_k^{\dagger}[\boldsymbol{R}_k]_{:,\mathcal{J}^c} = \boldsymbol{U}_k\boldsymbol{U}_k^{\dagger}[\boldsymbol{R}_k]_{:,\mathcal{J}^c},$$
$$[\boldsymbol{X}_k]_{\mathcal{I}^c,\mathcal{J}} = [\boldsymbol{C}_k]_{\mathcal{I}^c,:}\boldsymbol{U}_k^{\dagger}[\boldsymbol{R}_k]_{:,\mathcal{J}} = [\boldsymbol{C}_k]_{\mathcal{I}^c,:}\boldsymbol{U}_k^{\dagger}\boldsymbol{U}_k,$$
$$[\boldsymbol{X}_k]_{\mathcal{I},\mathcal{J}} = [\boldsymbol{C}_k]_{\mathcal{I},:}\boldsymbol{U}_k^{\dagger}[\boldsymbol{R}_k]_{:,\mathcal{J}} = \boldsymbol{U}_k. \tag{11}$$

The computational complexity of (11) is $\mathcal{O}(nr(|\mathcal{I}| + |\mathcal{J}|))$. Finally, computing the stopping criterion e_k, i.e., (10), costs $\mathcal{O}(|\Omega_{\boldsymbol{R}}| + |\Omega_{\boldsymbol{C}}|)$ flops.

Overall, ICURC costs total $\mathcal{O}(nr(|\mathcal{I}| + |\mathcal{J}|))$ flops per iteration. Moreover, (11) also suggests that we only need to pass the updated CUR components (i.e., $[\boldsymbol{R}_k]_{:,\mathcal{J}^c}$, $[\boldsymbol{C}_k]_{\mathcal{I}^c,:}$, and $\boldsymbol{U}_k$) to the next iteration instead of the much larger $\boldsymbol{X}_k$. Hence, we conclude that ICURC is computationally and memory efficient provided $|\mathcal{I}|, |\mathcal{J}| \ll n$.

Remark 3. Empirically, we observe that ICURC achieves a linear convergence rate when the samples are concentrated on a smaller collection of rows and columns. For the interested reader, we include several numerical results for ICURC's convergence behavior in the supplemental material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2023.3261185.

IV. Numerical Experiments

In this section, we compare the empirical performance of our CCS-model-based ICURC, i.e., Algorithm 2, against the state-of-the-art algorithms (ScaledPGD [32] and SVP [28]) based on uniform sampling method. The related codes are provided in https://github.com/huangl3/CCS-ICURC. All experiments are implemented on Matlab R2020a and executed on a Linux workstation equipped with Intel i9-9940X CPU (3.3 GHz @ 14 cores) and 128 GB DDR4 RAM. We emphasize again that our approach is computationally and memory efficient. All our experiments can be reproduced with a basic laptop.

A. Synthetic Examples

In matrix completion, an important question is how many measurements are needed for an algorithm to ensure a reliable reconstruction of a low-rank matrix. Thus, we investigate the recoverability of the ICURC algorithm based on the CCS model in the framework of phase transitions:

1) Phase transition that studies the required sampling rate over the whole matrix based on different sizes of the selected row and column submatrices and different uniform sampling rates on the selected submatrices.
2) Phase transition that explores the required measurements over different sizes of the original problem.

On the synthetic simulation, we only consider square problems and consider a low-rank matrix $\boldsymbol{X} \in \mathbb{R}^{n\times n}$ of rank r. Thus, we sample the same number of rows and columns in the following simulations, i.e., $|\mathcal{I}| = |\mathcal{J}|$. In comparison, we have also compared our results with that of SVP and ScaledPGD based on a uniform sampling model.

1) Empirical Phase Transition: First, we explore the recovery ability of ICURC for CCS with different combinations of the sampling number of rows and columns with $|\mathcal{I}| = |\mathcal{J}| = \delta n$ and the uniform sampling rate p on the selected rows and columns. This experiment runs on the matrix of size $10^3 \times 10^3$ and under different rank settings with rank $r \in \{5, 10, 15\}$. For each rank, we generate 20 test examples for each given pair of (δ, p) and an example is considered to be successfully solved if the relative error

$$\varepsilon_k := \frac{\|\boldsymbol{X} - \boldsymbol{C}_k\boldsymbol{U}_k^{\dagger}\boldsymbol{R}_k\|_{\mathrm{F}}}{\|\boldsymbol{X}\|_{\mathrm{F}}} \leq 10^{-2}. \tag{12}$$

These simulation results are summarized in Fig. 3, where the first row presents the 3D view of the phase transition results and the second row shows the corresponding 2D view results by adding the uniform sampling results in the last two columns. In the 3D result, the z-axis stands for the overall sampling rate over the whole matrix. In Fig. 3, a white pixel represents the successful completion of these 20 tests and a black pixel means all the 20 tests are failed. From these figures, one can see that as the rank r increases, the required overall sampling rate becomes larger to guarantee successful completion since a larger rank r leads to a more difficult problem. Moreover, regardless of the combinations of the sizes of the concentrated row and column submatrices and the sampling rates on the selected submatrices, we guarantee matrix completion as long as the combinations result in a sufficiently large total sampling rate (see the second row of Fig. 3). This observation shows that CCS provides the flexibility to sample low-rank matrix and still ensures the successful completion of the underlying low-rank matrix from its samples. From the second row of Fig. 3, one can see that the required sampling rates for ICURC on the CCS model are comparable with that for the ScaledPGD [32] and the SVP [28] algorithms on the uniform sampling.

We also investigate the recovery ability of our ICURC for CCS in the framework of phase transition by studying the relation between the required measurements of the underlying low-rank matrices and their size n. The results are reported in Fig. 4. Therein, we first uniformly sample a row submatrix $\boldsymbol{R} \in \mathbb{R}^{cr\log^2(n)\times n}$ and a column submatrix $\boldsymbol{C} \in \mathbb{R}^{n\times cr\log^2(n)}$, where $r \in \{5, 10, 15\}$ and $c \in \{0.25, 0.5, 1\}$. Then, we uniformly sample $s/2$ entries on each submatrix, i.e., s samples in total and the intersection submatrix $\boldsymbol{U}$ has a denser sampling rate than $\boldsymbol{R}$ and $\boldsymbol{C}$. Similar to Fig. 3, we generate 20 problems for each given pair of (n, s) under each setting of (r, c), and a problem is considered to be successfully solved if $\varepsilon_k \leq 10^{-2}$ (recall ε_k is defined in (12)). From Fig. 4, one can see that the overall required samples to guarantee the recovery of the missing data is independent of the size of the concentrated submatrices. These observations further illustrate the flexibility of our CCS model.

B. Image Recovery

In this section, we compare the matrix completion performances solved by ICURC using CCS and by ScaledPGD [32] and SVP [28] using uniform sampling for image recovery. The simulations are tested on two grey-scaled images, namely "Building"[4] and "Window"[5] of size 2000×3000 by recording

[4]https://pxhere.com/en/photo/57707.
[5]https://pxhere.com/en/photo/1421981.

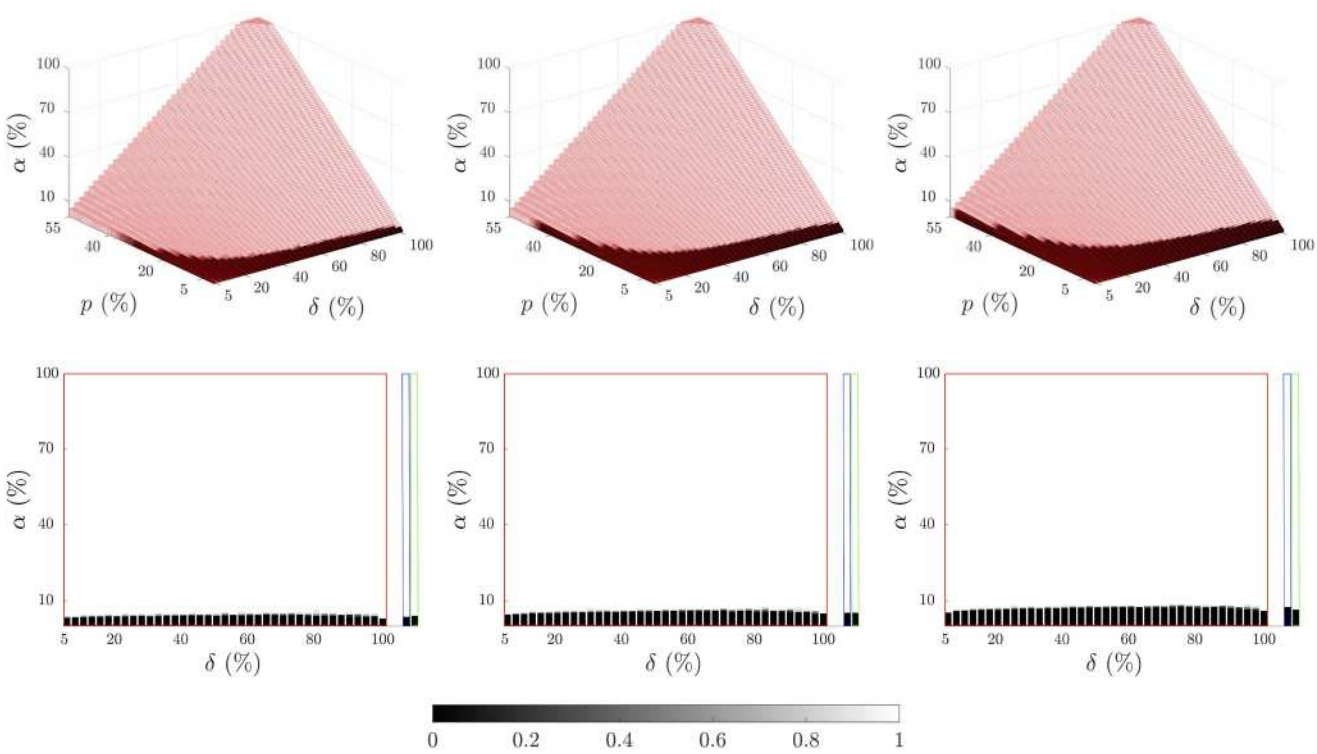


Fig. 3. Empirical phase transition in the overall sampling rate α, the percentage of selected rows and columns δ, and uniform sampling rates on the selected submatrices p. *Row 1*: 3D-view of the empirical phase transition of ICURC. *Row 2*: 2D view of empirical phase transition of ICURC (in the red box), ScaledPGD (in the blue box), and SVP (in the green box). *Left*: $r = 5$. *Middle*: $r = 10$. *Right*: $r = 15$. One can see that as rank increases, the required overall sampling rate increases correspondingly. Additionally, the CCS model provides flexibility in obtaining a sufficient amount of data to ensure completing the missing data successfully and the performance of the ICURC algorithm from the CCS-based samples is comparable to that of the state-of-the-art algorithms (SVP and ScaledPGD) from the uniform-sampling-based samples.

reconstruction quality and runtime. The reconstruction quality is measured by the signal-to-noise ratio (SNR), which is defined as

$$\mathrm{SNR}_{\mathrm{dB}}(\widetilde{\boldsymbol{X}}) = 20\log_{10}\left(\frac{\|\boldsymbol{X}\|_{\mathrm{F}}}{\|\widetilde{\boldsymbol{X}} - \boldsymbol{X}\|_{\mathrm{F}}}\right),$$

where $\boldsymbol{X}$ is the original image and $\widetilde{\boldsymbol{X}}$ represents the reconstructed image.

In this simulation, we aim to find a rank-20 approximation $\widetilde{\boldsymbol{X}}$ for the given image $\boldsymbol{X}$. First we generate the observations according to the CCS model. We randomly select the concentrated row and column submatrices $\boldsymbol{R}$ and $\boldsymbol{C}$ with row indices $\mathcal{I}$ of size δm and column indices $\mathcal{J}$ of size δn columns, i.e., $\boldsymbol{R} = [\boldsymbol{X}]_{\mathcal{I},:}$ and $\boldsymbol{C} = [\boldsymbol{X}]_{:,\mathcal{J}}$. Then, we randomly select $\frac{\alpha mn}{2}$ entries on each submatrix and denote the corresponding indices of the observed entries by $\Omega_{\boldsymbol{R}}$ and $\Omega_{\boldsymbol{C}}$, which result in two partially observed submatrices $\boldsymbol{R}_{\mathrm{obs}} = [\mathcal{P}_{\Omega_{\boldsymbol{R}}}(\boldsymbol{X})]_{\mathcal{I},:}$ and $\boldsymbol{C}_{\mathrm{obs}} = [\mathcal{P}_{\Omega_{\boldsymbol{C}}}(\boldsymbol{X})]_{:,\mathcal{J}}$. As a result, we obtain a partially observed image whose observed entries are concentrated on $\boldsymbol{R}$ and $\boldsymbol{C}$. Then we fill in the missing pixels by applying our ICUR algorithm. In comparison, we also generate αmn observations based on the uniform sampling model over the original matrix X. After that, we fill in the missing data via ScaledPGD or SVP. The above processes are repeated for 10 times.

The averaged test results on different α (i.e., overall observation rates) are summarized in Table II. Meanwhile, we provide some visual results in Fig. 5. It shows that all the algorithms achieve visually reliable results. Additionally, in comparison with the visual results in Fig. 2, our ICURC algorithm has a much better performance than SVP and ScaledPGD algorithms in solving the image inpainting problems when the observed pixels are selected based on our CCS model. From Table II, one can observe that regardless of different combinations of the sizes of the concentrated row and column submatrices and the sampling rates, the qualities of the results from ICURC are similar as long as the overall sampling rates are the same. This observation further illustrates the flexibility of the CCS model. Additionally, one can also find that ICURC on CCS achieves comparable (even better) quality with the algorithms on the uniform sampling model. From the runtime perspective, our ICURC on the CCS model is substantially faster than ScaledPGD and SVP on uniform sampling.

C. Recommendation System

Recommendation systems aim to predict the users' preferences from partial information of personalized item recommendations. Each dataset in a recommendation system can be represented as a matrix by arranging each item's ratings as a row and each user's ratings as a column. If we view the unobserved

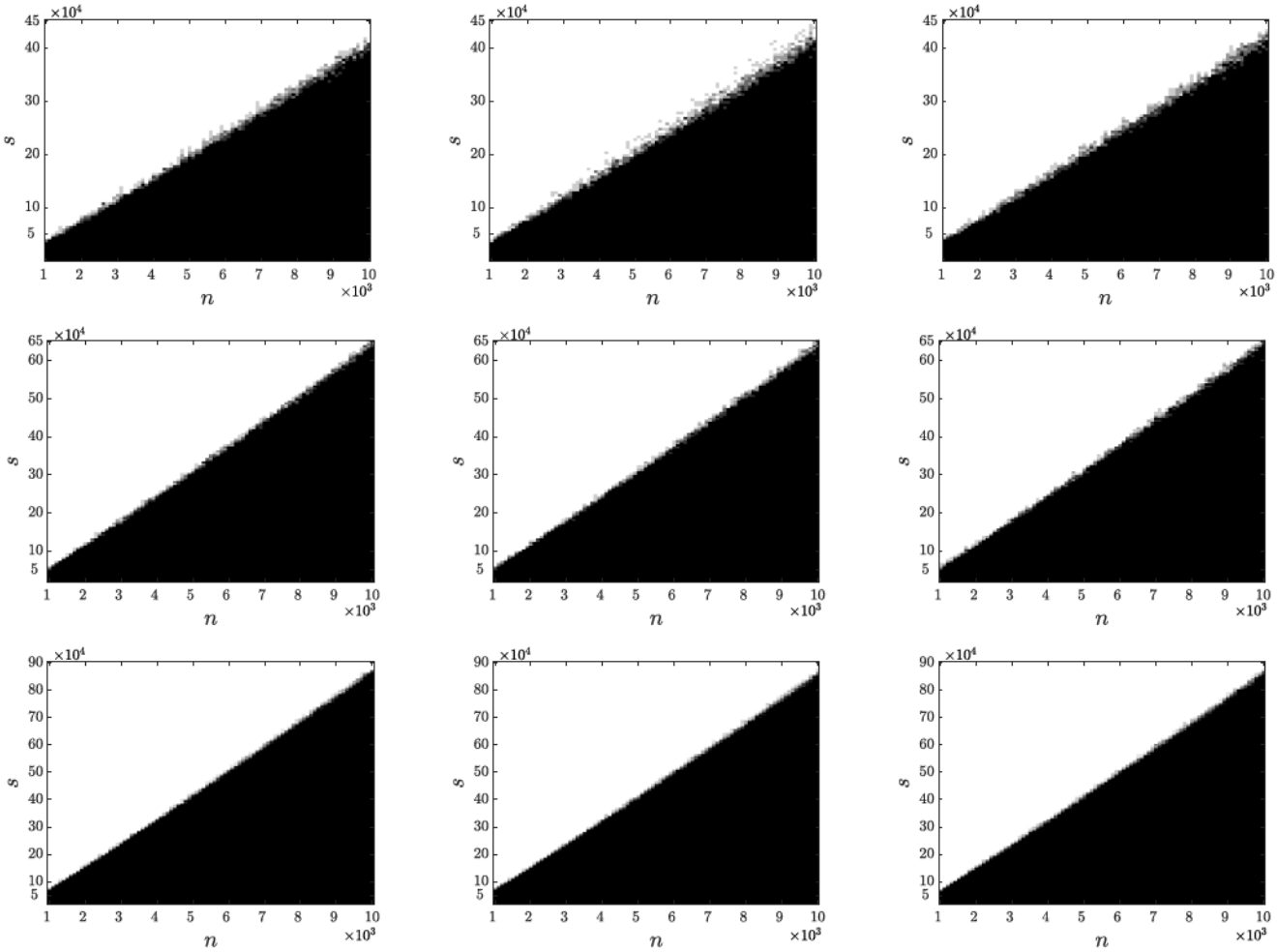


Fig. 4. Empirical phase transitions of ICURC in overall observation size s and problem size n. The column (resp. row) number of the concentrated column (resp. row) submatrix equals to $cr\log^2(n)$. *Row 1*: $r = 5$. *Row 2*: $r = 10$. *Row 3*: $r = 15$. *Left*: $c = 0.25$. *Middle*: $c = 0.5$. *Right*: $c = 1$. The required samples for guaranteed matrix completion are independent of the size of the concentrated submatrices.

TABLE II
IMAGE INPAINTING RESULTS ON BUILDING AND WINDOW DATASETS. UNDER VARIOUS SETUPS OF CCS, ICURC CAN ACHIEVE HIGHER SNR WITH SHORTER RUNTIME COMPARED WITH OTHER METHODS

DATASET		Building			Window		
OVERALL OBSERVATION RATE (α)		10 %	12 %	14 %	10 %	12 %	14 %
SNR	ICURC-8	**23.762**	**24.750**	**25.195**	31.792	32.343	32.5533
	ICURC-9	23.688	24.557	25.108	**31.831**	**32.513**	**32.984**
	ICURC-10	23.629	24.229	24.823	31.546	32.364	32.911
	ScaledPGD	20.593	21.722	21.734	31.338	31.918	32.693
	SVP	18.065	18.940	19.607	27.451	29.4900	30.8541
RUNTIME (sec)	ICURC-8	**0.400**	**0.366**	**0.306**	**0.339**	**0.244**	**0.313**
	ICURC-9	0.482	0.416	0.395	0.343	0.313	0.364
	ICURC-10	0.616	0.518	0.425	0.627	0.462	0.396
	ScaledPGD	3.518	3.405	3.230	2.722	2.314	2.141
	SVP	9.925	14.675	14.661	10.626	10.102	9.873

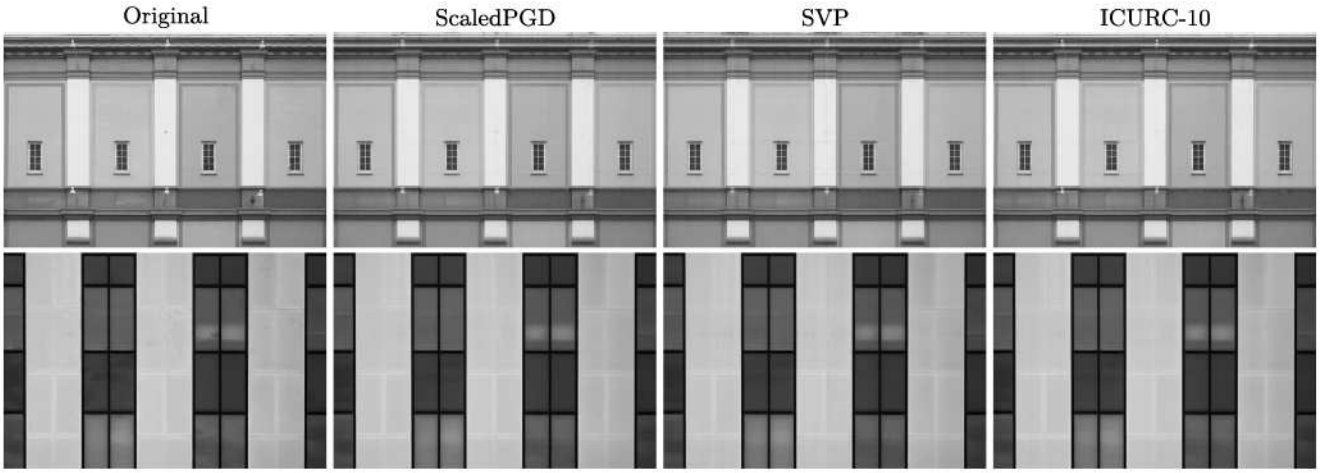


Fig. 5. Visual results for image inpainting by setting rank $r = 20$ and the percentage of selected rows and columns $\delta = 10\%$. ScaledPGD and SVP are based on the uniform sampling model with the same observed number of entries as the one based on CCS. All algorithms achieve visually reliable results.

TABLE III
DATASETS INFORMATION FOR COLLABORATIVE FILTERING. HERE, α IS THE OVERALL OBSERVATION RATE

DATASET	#USERS	#ITEMS	α (%)	RATING RANGE
ML-100K	943	1682	4.190	1–5
ML-10M	2000	3000	32.92	1–10
FilmTrust	1508	2071	2.660	1–8

ratings as missing entries of data matrices, predicting the missing ratings can be considered a matrix completion problem, as the underlying matrix is expected to be low-rank since only a few factors contribute to an individual's preferences [46]. In this section, we evaluate the performance of our CCS model solving by ICURC algorithm on three datasets namely the Movie-1 M, the Movie-10 M datasets from the Movielens research project[6] [47] and the FilmTrust dataset[7] [48]. For a given dataset, we first generate an item-user matrix of size $m \times n$. Due to the large size and low observation rate of the Movie-10 M dataset, we follow the instructions in [49] to extract a 2000×3000 submatrix based on the observation rates on rows and columns. The characteristics of all the data used in our simulations are summarized in Table III.

To evaluate the performance, we employ the Cross-Validation method. For each run, the observed data is randomly split into training and testing sets denoted by Ω_{train} and Ω_{test} [50]. More specifically, we randomly select δm rows and δn columns and then randomly choose αmn entries from the observed entries on the selected rows and columns to form $\Omega^{(1)}_{\text{train}}$ for CCS model. For comparison, we also randomly choose αmn entries from the observed data over the whole matrix to form a new training dataset $\Omega^{(2)}_{\text{train}}$ based on the uniform sampling model. The information on the training and testing sets for each data matrix is summarized in Table III. After the training and testing datasets are generated, we run the ICURC algorithm on $\Omega^{(1)}_{\text{train}}$, and ScaledPGD and SVP algorithms on $\Omega^{(2)}_{\text{train}}$.

Following [50], we adopt two different methods to measure the recommendation quality on the testing datasets. The first one is the hit-rate (HR) which is defined as the ratio of the number of hits to the size of the testing dataset:

$$\text{HR} = \frac{\#\text{hits}}{|\Omega_{\text{test}}|}, \tag{13}$$

where a predicted rating P_i is considered as a hit if its rounded value is equal to the actual rating A_i in the test set. To penalize each missed prediction and to emphasize the errors, we also computed the Normalized Mean Absolute Error (NMAE) [51], [52] defined as follows:

$$\text{NMAE} = \frac{1}{|\Omega_{\text{test}}|(S_{\max} - S_{\min})} \sum_{i \in \Omega_{\text{test}}} |P_i - A_i|, \tag{14}$$

where $S_{\max}$ and $S_{\min}$ denote the maximum and minimum rating, respectively.

Fig. 6 summaries the averaged numerical results over 10 independent trials. One can see that the results for the ICURC algorithm based on the CCS model have better performance compared with the ones for ScaledPGD and SVP algorithms on the uniform sampling model. Specifically, ICURC can reach higher HR and lower NMAE in a much shorter runtime compared with other methods. This further illustrates that ICURC is computationally efficient. Fixing the overall sampling rate for the CCS model, one can observe that the performances for different combinations of the concentrated row and column submatrices are comparable. This observation further illustrates the flexibility of our CCS model.

D. Link Prediction

In link prediction problems, we are given a graph $G = (V, E)$, that has vertices V and edges E, represented in an adjacency matrix A. If there exists an observed link between vertices i and j, then $A_{i,j} = 1$; otherwise $A_{i,j} = 0$. Link prediction

[6]Movie-100 K and the Movie-10 M datasets can be downloaded from https://grouplens.org/datasets/movielens.

[7]FilmTrust dataset can be found at https://guoguibing.github.io/librec/datasets.html.

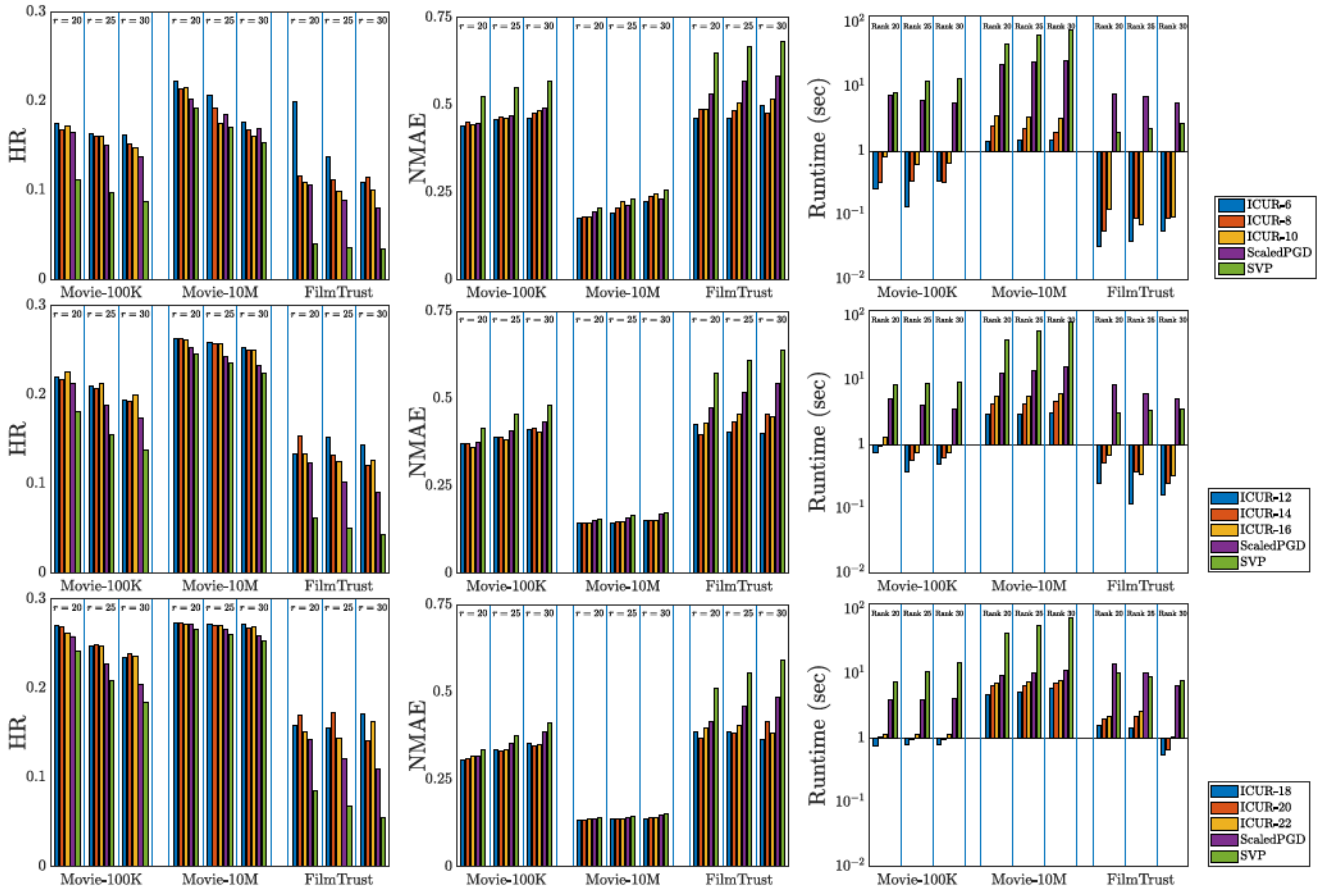


Fig. 6. Bar-plot results on the recommendation system data. The performances are measured in HR, NMAE, and runtime. *Top*: overall observation rate $\alpha = 10\%$. *Middle*: $\alpha = 20\%$. *Bottom*: $\alpha = 30\%$. When overall observation rate α is fixed, the ICURC performs better under various CCS conditions compared with other methods in uniform sampling.

problem aims to learn the distribution of existing links and, thus, to predict the potential links in the graph [53]. In this section, we evaluate the performance of our CCS model solved by the ICURC algorithm on three link prediction datasets namely Blogs[8] containing the hyperlinks between blogs in the context of 2004 US election, Opsahl[9] indicating flights between airports around the world, and Figeys[10] describing interactions between proteins. The three datasets come from the Koblenz Network Collection (KONECT [54]). The characteristics of these datasets are summarized in Table IV.

To evaluate the performance of our methods, we follow the works of [53], [55] by randomly dividing the existing links into training and testing samples. From the perspective of matrix completion, this is the same as randomly splitting the observed entries of the adjacency matrix A into corresponding Ω_{train} and Ω_{test}. Similar to the setup of the recommendation system problem in Section IV-C, we generate different training datasets based on the CCS model and non-CCS model and denote them as $\Omega^{(1)}_{\text{train}}$ and $\Omega^{(2)}_{\text{train}}$ respectively. We apply ICURC algorithm on $\Omega^{(1)}_{\text{train}}$ and ScaledPGD and SVP algorithms on $\Omega^{(2)}_{\text{train}}$. We also adopt two popular metrics, Precision [56], which focuses on the top predicted links, and Area Under the receiver operating characteristic Curve (AUC) h [57], which evaluates the entire set of predicted links. Precision is defined as the ratio of the actual number of connected edges to the predicted number of connected edges. Links predicted by algorithms can be interpreted as the likelihood of unobserved (new) links; the higher likelihood indicates a greater possibility of an unobserved link [53]. We sort the entries of predicted links in descending order and select the top L links. L is chosen to be the cardinality of the testing dataset [53]. Let L_{m} be the number of links in the top L predicted links that appear in the testing dataset, Precision can be calculated

TABLE IV
INFORMATION FOR LINK PREDICTION DATASETS. HERE, α IS THE OVERALL OBSERVATION RATE

DATASET	SIZE (#NODES)	α (%)
Blogs	1224	1.269
Opsahl	2939	0.353
Figeys	2239	0.357

[8]Blogs dataset can be found in http://konect.cc/networks/moreno_blogs.

[9]Opsahl dataset can be found in http://konect.cc/networks/opsahl-openflights.

[10]Figeys dataset can be found in http://konect.cc/networks/maayan-figeys.

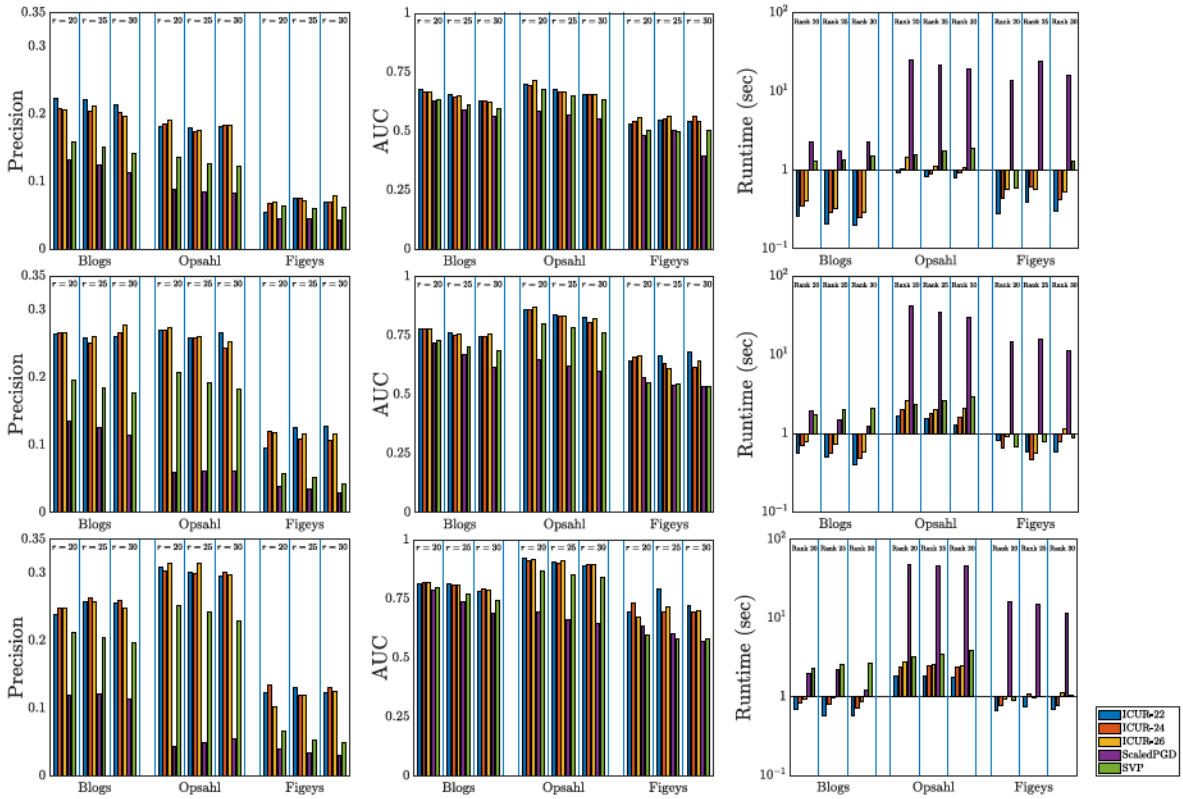


Fig. 7. Bar-plot results on link prediction datasets. The performances are measured in Precision, AUC, and runtime. *Top*: overall observation rate $\alpha = 10\%$. *Middle*: $\alpha = 20\%$. *Bottom*: $\alpha = 30\%$. When overall observation rate α is fixed, the ICURC performs better under various CCS conditions compared with other methods in uniform sampling.

by

$$\text{Precision} = \frac{L_{\text{m}}}{L}, \tag{15}$$

where the higher Precision is, the more accurate of the prediction is [56].

AUC measures the area under the receiver operating characteristic curve, which can be interpreted as the probability that a randomly chosen missing link from the set of predicted Ω_{test} is given a higher likelihood than a randomly chosen potentially non-existing link (which is the set of all unobserved links from G) [57], [58]. The AUC is calculated as:

$$\text{AUC} = \frac{y_{\text{m}} + 0.5y_{\text{n}}}{y}, \tag{16}$$

where y is the number of independent comparisons between each randomly picked pair of a missing link and a non-existing link. y_{m} is the times that the missing links have a higher predicted likelihood than non-existing links while y_{n} counts the number of times if their likelihoods are equal. We use $y = 5000$. The degree to which the AUC exceeds 0.5 illustrates how much better the predictions are compared with a random guess [55].

Fig. 7 summarises the averaged numerical results over 10 independent trials for each fixed r, α, and δ. One can see that based on the CCS model, ICURC performs better compared with the ones based on the uniform sampling model solved by other methods. Particularly, when the overall sampling rate is fixed, the ICURC based on different concentrated rows and columns can reach higher Precision and AUC under shorter intervals. This, again, confirms the computational efficiency of ICURC and the flexibility of our CCS model.

V. Proofs

In this section, we provide the proofs for the theoretical results presented in Section II-A, i.e., Lemma 3 and Theorem 4.

A. *Proof of Lemma 3*

Proof of Lemma 3 We invoke [24, Theorem 3.5]. By setting $\delta = 0.75$ and $\gamma = r\log^2(n)$ in that theorem, the following inequalities hold

$$\sqrt{\frac{|\mathcal{I}|}{n}}\left\|[\boldsymbol{W}]^{\dagger}_{\mathcal{I},:}\right\|_2 \leq 2,$$
$$\mu_{1\boldsymbol{R}} \leq 4\kappa^2\mu_1,$$
$$\mu_{2\boldsymbol{R}} \leq \mu_2,$$
$$\kappa_{\boldsymbol{C}} \leq 2\sqrt{\mu_1 r}\kappa,$$

with probability at least

$$1 - \frac{r}{\exp((0.75 + 0.25\log(0.25))r\log^2(n))} \geq 1 - \frac{r}{n^{0.4r\log(n)}}.$$

This completes the proof. ■

Algorithm 3: Two-Step Completion (TSC) for CCS.

1: **Input**: $[\boldsymbol{X}]_{\Omega_{\boldsymbol{R}}\cup\Omega_{\boldsymbol{C}}}$: observed data; $\Omega_{\boldsymbol{R}}, \Omega_{\boldsymbol{C}}$: observation locations; $\mathcal{I}, \mathcal{J}$: row and column indices that define $\boldsymbol{R}$ and $\boldsymbol{C}$ respectively; r: target rank; MC: the chosen matrix completion solver.
2: $\widetilde{\boldsymbol{R}} = \text{MC}([\boldsymbol{X}]_{\Omega_{\boldsymbol{R}}}, r)$
3: $\widetilde{\boldsymbol{C}} = \text{MC}([\boldsymbol{X}]_{\Omega_{\boldsymbol{C}}}, r)$
4: $\widetilde{\boldsymbol{U}} = \widetilde{\boldsymbol{C}}(\mathcal{I}, :)$
5: $\widetilde{\boldsymbol{X}} = \widetilde{\boldsymbol{C}}\widetilde{\boldsymbol{U}}^{\dagger}\widetilde{\boldsymbol{R}}$
6: **Output:** $\widetilde{\boldsymbol{X}}$: approximation of $\boldsymbol{X}$.

B. Proof of Theorem 4

The proof of our main theorem (i.e., Theorem 4) is based on our Two-Step Completion (TSC) algorithm. For the ease of readers, we state the TSC algorithm first.

Proof of Theorem 4 Since $\mathcal{I}$ and $\mathcal{J}$ are chosen uniformly from $[n]$, according to Lemma 3 we have that

$$\mu_{2\boldsymbol{C}} \le 4\kappa^2\mu_2, \quad \mu_{1\boldsymbol{R}} \le 4\kappa^2\mu_1,$$
$$\kappa_{\boldsymbol{C}} \le 2\sqrt{\mu_2 r}\kappa, \quad \kappa_{\boldsymbol{R}} \le 2\sqrt{\mu_2 r}\kappa,$$

with probability at least $1 - \frac{2r}{n^{0.4r\log(n)}}$. Thus,

$$\|\boldsymbol{W}_{\boldsymbol{C}}\boldsymbol{V}_{\boldsymbol{C}}^{\top}\|_{\infty} \le 2\kappa\sqrt{r\mu_1\mu_2}\sqrt{\frac{r}{n|J|}}$$

and

$$\|\boldsymbol{W}_{\boldsymbol{R}}\boldsymbol{V}_{\boldsymbol{R}}^{\top}\|_{\infty} \le 2\kappa\sqrt{r\mu_1\mu_2}\sqrt{\frac{r}{n|I|}}$$

hold with probability at least $1 - \frac{2r}{n^{0.4r\log(n)}}$.

By [13, Theorem 2], the following two statements hold:

1) $|\Omega_{\boldsymbol{C}}| \ge 128\kappa^2 r^2\mu_1\mu_2(n+|\mathcal{J}|)\beta\log^2(n)$ for some $\beta > 1$ ensures that $\boldsymbol{C}$ is the minimizer to the problem

$$\underset{\widetilde{\boldsymbol{C}}}{\text{minimize}} \quad \|\widetilde{\boldsymbol{C}}\|_*$$
$$\text{subject to} \quad \mathcal{P}_{\Omega_{\boldsymbol{C}}}(\widetilde{\boldsymbol{C}}) = \mathcal{P}_{\Omega_{\boldsymbol{C}}}(\boldsymbol{C})$$

with probability at least

$$1 - \frac{6\log(n)}{(n+\mu_2 r^2\log^2(n))^{2\beta-2}} - \frac{1}{n^{2\beta^{0.5}-2}}.$$

2) $|\Omega_{\boldsymbol{R}}| \ge 128\kappa^2 r^2\mu_1\mu_2(n+|\mathcal{I}|)\beta\log^2(n)$ for some $\beta > 1$ ensures that $\boldsymbol{R}$ is the minimizer to the problem

$$\underset{\widetilde{\boldsymbol{R}}}{\text{minimize}} \quad \|\widetilde{\boldsymbol{R}}\|_*$$
$$\text{subject to} \quad \mathcal{P}_{\Omega_{\boldsymbol{R}}}(\widetilde{\boldsymbol{R}}) = \mathcal{P}_{\Omega_{\boldsymbol{R}}}(\boldsymbol{R})$$

with probability at least

$$1 - \frac{6\log(n)}{(n+\mu_2 r^2\log^2(n))^{2\beta-2}} - \frac{1}{n^{2\beta^{0.5}-2}}.$$

Since $\kappa_{\boldsymbol{C}} \le 2\sqrt{\mu_2 r}\kappa$ and $\kappa_{\boldsymbol{R}} \le 2\sqrt{\mu_2 r}\kappa$, we thus have

$$\text{rank}(\boldsymbol{C}) = \text{rank}(\boldsymbol{R}) = \text{rank}(\boldsymbol{X}) = r.$$

According to [22, Theorem 5.5], we thus have $\boldsymbol{X} = \boldsymbol{C}\boldsymbol{U}^{\dagger}\boldsymbol{R}$, in other words, the CUR decomposition $\boldsymbol{C}\boldsymbol{U}^{\dagger}\boldsymbol{R}$ can reproduce the original $\boldsymbol{X}$.

Combining all the statements above, $\boldsymbol{X}$ can be exactly recovered from $\Omega_{\boldsymbol{C}} \cup \Omega_{\boldsymbol{R}}$ with probability at least

$$1 - \frac{2r}{n^{0.4r\log(n)}} - \frac{2}{n^{2\beta^{0.5}-2}} - \sum_{i=1}^{2}\frac{6\log(n)}{(n+\mu_i r^2\log^2(n))^{2\beta-2}}.$$

This completes the proof. ■

VI. Conclusion and Future Directions

This paper proposes a novel, easy-to-implement, and practically flexible sampling model, coined Cross-Concentrated Sampling (CCS), for matrix completion problems that bridges the classical uniform sampling model and the CUR sampling model. For this model, we provide a sufficient sampling bound to ensure the uniqueness of the solution for this matrix completion problem. Furthermore, we develop an efficient non-convex algorithm to solve the CCS-based MC. The efficiency of the algorithm is illustrated on synthetic and real datasets. The simulations also show that CCS provides flexibility to acquire sufficient data to ensure the successful completion that can potentially save costs in some real applications.

There are four lines for future work. First, the sufficient bound in this paper is not tight. One can see that if the CUR sampling model is applied to a low-rank matrix of size $n \times n$, $\mathcal{O}(nr\log(n))$ samples can ensure the successful completion with a high probability. There is room to improve the sampling bound for our CCS model. Second, it would be helpful to analyze the convergence guarantee of ICURC in future work. Third, our empirical simulations have shown the promising performance of ICURC, but there is likely still room for improvement. More efficient approaches will be developed with the theoretical foundation as a guide. Lastly, this novel sampling model is based on matrix CUR decomposition. Recently, tensor CUR decompositions have been proposed (cf. [59], [60]). In low-rank tensor approximation, the original tensor can be well-approximated by making good use of merely one smaller-sized subtensor and a small collection of fibers from each mode. Therefore, there is no need to access the full tensor, which makes tensor CUR decompositions memory and computationally efficient. We plan to generalize the proposed sampling model into this tensor setting.

References

[1] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, pp. 717–772, 2009.
[2] J. Bennett, C. Elkan, B. Liu, P. Smyth, and D. Tikk, "KDD cup and workshop 2007," *SIGKDD Explorations Newslett.*, vol. 9, no. 2, pp. 51–52, 2007.
[3] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Commun. ACM*, vol. 35, no. 12, pp. 61–70, 1992.
[4] P. Chen and D. Suter, "Recovering the missing components in a large noisy low-rank matrix: Application to SFM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 8, pp. 1051–1063, Aug. 2004.
[5] Y. Hu, D. Zhang, J. Ye, X. Li, and X. He, "Fast and accurate matrix completion via truncated nuclear norm regularization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 9, pp. 2117–2130, Sep. 2013.

[6] J.-F. Cai, T. Wang, and K. Wei, "Fast and provable algorithms for spectrally sparse signal reconstruction via low-rank Hankel matrix completion," *Appl. Comput. Harmon. Anal.*, vol. 46, no. 1, pp. 94–121, 2019.

[7] H. Cai, J.-F. Cai, and J. You, "Structured gradient descent for fast robust low-rank Hankel matrix completion," 2022, *arXiv:2204.03316*.

[8] E. C. Chi, H. Zhou, G. K. Chen, D. O. Del Vecchyo, and K. Lange, "Genotype imputation via matrix completion," *Genome Res.*, vol. 23, no. 3, pp. 509–518, 2013.

[9] T. Cai, T. T. Cai, and A. Zhang, "Structured matrix completion with applications to genomic data integration," *J. Amer. Statist. Assoc.*, vol. 111, no. 514, pp. 621–633, 2016.

[10] A. Argyriou, T. Evgeniou, and M. Pontil, "Convex multi-task feature learning," *Mach. Learn.*, vol. 73, no. 3, pp. 243–272, 2008.

[11] Z. Liu and L. Vandenberghe, "Interior-point method for nuclear norm approximation with application to system identification," *SIAM J. Matrix Anal. Appl.*, vol. 31, no. 3, pp. 1235–1256, 2010.

[12] A. Singer and M. Cucuringu, "Uniqueness of low-rank matrix completion by rigidity theory," *SIAM J. Matrix Anal. Appl.*, vol. 31, no. 4, pp. 1621–1641, 2010.

[13] B. Recht, "A simpler approach to matrix completion," *J. Mach. Learn. Res.*, vol. 12, no. 12, pp. 3413–3430, 2011.

[14] O. Klopp, "Noisy low-rank matrix completion with general sampling distribution," *Bernoulli*, vol. 20, no. 1, pp. 282–303, 2014.

[15] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, 2010.

[16] E. J. Candès and T. Tao, "The power of convex relaxation: Near-optimal matrix completion," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2053–2080, May 2010.

[17] R. H. Keshavan, A. Montanari, and S. Oh, "Matrix completion from a few entries," *IEEE Trans. Inf. Theory*, vol. 56, no. 6, pp. 2980–2998, Jun. 2010.

[18] P. Jain, P. Netrapalli, and S. Sanghavi, "Low-rank matrix completion using alternating minimization," in *Proc. 45th Annu. ACM Symp. Theory Comput.*, 2013, pp. 665–674.

[19] R. Sun and Z.-Q. Luo, "Guaranteed matrix completion via non-convex factorization," *IEEE Trans. Inf. Theory*, vol. 62, no. 11, pp. 6535–6579, Nov. 2016.

[20] N. Zamarashkin, "Pseudo-skeleton approximations by matrices of maximal volume," *Math. Notes*, vol. 62, pp. 515–519, 1997.

[21] J. Chiu and L. Demanet, "Sublinear randomized algorithms for skeleton decompositions," *SIAM J. Matrix Anal. Appl.*, vol. 34, no. 3, pp. 1361–1383, 2013.

[22] K. Hamm and L. Huang, "Perspectives on CUR decompositions," *Appl. Comput. Harmon. Anal.*, vol. 48, no. 3, pp. 1088–1099, 2020.

[23] H. Cai, K. Hamm, L. Huang, J. Li, and T. Wang, "Rapid robust principal component analysis: CUR accelerated inexact low rank estimation," *IEEE Signal Process. Lett.*, vol. 28, pp. 116–120, 2021.

[24] H. Cai, K. Hamm, L. Huang, and D. Needell, "Robust CUR decomposition: Theory and imaging applications," *SIAM J. Imag. Sci.*, vol. 14, no. 4, pp. 1472–1503, 2021.

[25] M. Xu, R. Jin, and Z.-H. Zhou, "CUR algorithm for partially observed matrices," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1412–1421.

[26] Y. Chen, "Incoherence-optimal matrix completion," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2909–2923, May 2015.

[27] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Rev.*, vol. 38, no. 1, pp. 49–95, 1996.

[28] P. Jain, R. Meka, and I. Dhillon, "Guaranteed rank minimization via singular value projection," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2010, pp. 937–945.

[29] P. Jain and P. Netrapalli, "Fast exact matrix completion with finite samples," in *Proc. Conf. Learn. Theory*, 2015, pp. 1007–1034.

[30] R. H. Keshavan, *Efficient Algorithms for Collaborative Filtering*. Stanford, CA, USA: Stanford Univ. Press, 2012.

[31] Q. Zheng and J. Lafferty, "Convergence analysis for rectangular matrix completion using Burer-Monteiro factorization and gradient descent," 2016, *arXiv:1605.07051*.

[32] T. Tong, C. Ma, and Y. Chi, "Accelerating ill-conditioned low-rank matrix estimation via scaled gradient descent," *J. Mach. Learn. Res.*, vol. 22, no. 150, pp. 1–63, 2021.

[33] B. Vandereycken, "Low-rank matrix completion by Riemannian optimization," *SIAM J. Optim.*, vol. 23, no. 2, pp. 1214–1236, 2013.

[34] K. Wei, J.-F. Cai, T. F. Chan, and S. Leung, "Guarantees of Riemannian optimization for low rank matrix completion," *Inverse Problems Imag.*, vol. 14, no. 2, 2020, Art. no. 233.

[35] H. Avron and C. Boutsidis, "Faster subset selection for matrices and applications," *SIAM J. Matrix Anal. Appl.*, vol. 34, no. 4, pp. 1464–1499, 2013.

[36] A. Bhaskara, A. Rostamizadeh, J. Altschuler, M. Zadimoghaddam, T. Fu, and V. Mirrokni, "Greedy column subset selection: New bounds and distributed algorithms," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 2539–2548.

[37] X. Li and Y. Pang, "Deterministic column-based matrix decomposition," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 1, pp. 145–149, Jan. 2010.

[38] P. Drineas, R. Kannan, and M. W. Mahoney, "Fast Monte Carlo algorithms for matrices III: Computing a compressed approximate matrix decomposition," *SIAM J. Comput.*, vol. 36, no. 1, pp. 184–206, 2006.

[39] P. Drineas, M. W. Mahoney, and S. Muthukrishnan, "Relative-error CUR matrix decompositions," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 2, pp. 844–881, 2008.

[40] K. Hamm and L. Huang, "Stability of sampling for CUR decompositions," *Found. Data Sci.*, vol. 2, no. 2, 2020, Art. no. 83.

[41] M. W. Mahoney and P. Drineas, "CUR matrix decompositions for improved data analysis," *Proc. Nat. Acad. Sci. USA*, vol. 106, no. 3, pp. 697–702, 2009.

[42] S. Wang and Z. Zhang, "Improving CUR matrix decomposition and the Nyström approximation via adaptive sampling," *J. Mach. Learn. Res.*, vol. 14, pp. 2729–2769, 2013.

[43] J. A. Tropp, "Column subset selection, matrix factorization, and eigenvalue optimization," in *Proc. 20th Annu. ACM-SIAM Symp. Discrete Algorithms*, SIAM, 2009, pp. 978–986.

[44] K. Hamm and L. Huang, "Perturbations of CUR decompositions," *SIAM J. Matrix Anal. Appl.*, vol. 42, no. 1, pp. 351–375, 2021.

[45] K. Hamm, M. Meskini, and H. Cai, "Riemannian CUR decompositions for robust principal component analysis," in *Proc. Topological Algebr. Geometric Learn. Workshop*, 2022, pp. 152–160.

[46] Y. Plan, "Compressed sensing, sparse approximation, and low-rank matrix estimation," Ph.D. thesis, California Inst. Technol., California, USA, 2011. [Online]. Available: https://www.proquest.com/dissertations-theses/compressed-sensing-sparse-approximation-low-rank/docview/1020104518/

[47] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, 2015, Art. no. 19.

[48] G. Guo, J. Zhang, and N. Yorke-Smith, "A novel Bayesian similarity measure for recommender systems," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 2619–2625.

[49] V. Kalofolias, X. Bresson, M. Bronstein, and P. Vandergheynst, "Matrix completion on graphs," 2014, *arXiv:1408.1717*.

[50] H. Yildirim and M. S. Krishnamoorthy, "A random walk method for alleviating the sparsity problem in collaborative filtering," in *Proc. ACM Conf. Recommender Syst.*, 2008, pp. 131–138.

[51] S. Ma, D. Goldfarb, and L. Chen, "Fixed point and Bregman iterative methods for matrix rank minimization," *Math. Prog.*, vol. 128, no. 1, pp. 321–353, 2011.

[52] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, "Eigentaste: A constant time collaborative filtering algorithm," *Inf. Retrieval*, vol. 4, no. 2, pp. 133–151, 2001.

[53] R. Pech, D. Hao, L. Pan, H. Cheng, and T. Zhou, "Link prediction via matrix completion," *Europhysics Lett.*, vol. 117, no. 3, 2017, Art. no. 38002.

[54] J. Kunegis, "KONECT – The Koblenz network collection," in *Proc. Int. Conf. World Wide Web Companion*, 2013, pp. 1343–1350.

[55] F. Gao, K. Musial, C. Cooper, and S. Tsoka, "Link prediction methods and their accuracy for different social networks and network metrics," *Sci. Program.*, vol. 2015, 2015, Art. no. 172879.

[56] Z. Zhao, Z. Gou, Y. Du, J. Ma, T. Li, and R. Zhang, "A novel link prediction algorithm based on inductive matrix completion," *Expert Syst. Appl.*, vol. 188, 2022, Art. no. 116033.

[57] A. Clauset, C. Moore, and M. E. Newman, "Hierarchical structure and the prediction of missing links in networks," *Nature*, vol. 453, no. 7191, pp. 98–101, 2008.

[58] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve," *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[59] H. Cai, K. Hamm, L. Huang, and D. Needell, "Mode-wise tensor decompositions: Multi-dimensional generalizations of CUR decompositions," *J. Mach. Learn. Res.*, vol. 22, no. 185, pp. 1–36, 2021.

[60] H. Cai, Z. Chao, L. Huang, and D. Needell, "Fast robust tensor principal component analysis via fiber CUR decomposition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops*, 2021, pp. 189–197.

HanQin Cai received the PhD degree in applied mathematics and computational sciences from the University of Iowa. He is currently the Paul N. Somerville Endowed assistant professor with the Department of Statistics and Data Science and the Department of Computer Science, University of Central Florida. He is also the director of Data Science Lab. His research interests include machine learning, data science, mathematical optimization, and applied harmonic analysis.

Pengyu Li received the BS degrees in applied mathematics (with Daus Prize) and statistics from the University of California, Los Angeles. He is currently working toward the PhD degree with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor. His research interests include matrix analysis, deep neural network, data science, and numerical linear algebra.

Longxiu Huang received the PhD degree in applied mathematics from Vanderbilt University. She is currently an assistant professor with the Department of Computational Mathematics, Science, and Engineering and the Department of Mathematics, Michigan State University. Her research interests include applied harmonic analysis, machine learning, data science, numerical linear algebra, and approximation theory.

Deanna Needell received the PhD degree in mathematics from the University of California, Davis. She is currently a professor with the Department of Mathematics, University of California, Los Angeles. She recently won the IMA Prize and is a 2022 AMS fellow. Her research interests include applied harmonic analysis, machine learning, data science, numerical linear algebra, and applications in data science and equity.