On the Optimal Load-Memory Tradeoff of Cache-Aided Scalar Linear Function Retrieval

Kai Wan, *Member, IEEE*, Hua Sun, *Member, IEEE*, Mingyue Ji, *Member, IEEE*, Daniela Tuninetti, *Fellow, IEEE*, and Giuseppe Caire, *Fellow, IEEE*

***Abstract*— Coded caching has the potential to greatly reduce network traffic by leveraging the cheap and abundant storage available in end-user devices so as to create multicast opportunities in the delivery phase. In the seminal work by Maddah-Ali and Niesen (MAN), the shared-link coded caching problem was formulated, where each user demands one file (i.e., single file retrieval). This article generalizes the MAN caching problem formulation from single file retrieval on the binary filed to general *scalar linear function retrieval* on an arbitrary finite field. The proposed novel scheme is linear, based on MAN uncoded cache placement, and leverages ideas from interference alignment. Quite surprisingly, the worst-case load of the proposed scheme among all possible demands is the same as the one of the scheme by Yu, Maddah-Ali, and Avestimehr (YMA) for single file retrieval. The proposed scheme has thus the same optimality guarantees as YMA, namely, it is optimal under the constraint of uncoded cache placement, and is optimal to within a factor 2 otherwise. Some extensions of the proposed scheme are then discussed. It is shown that the proposed scheme works not only on arbitrary finite field, but also on any commutative ring. The key idea of this article can be also extended to all scenarios to which the original MAN scheme has been extended, including but not limited to demand-private retrieval and Device-to-Device networks.**

***Index Terms*— Coded caching, uncoded cache placement, scalar linear function retrieval.**

Manuscript received January 12, 2020; revised October 26, 2020; accepted March 8, 2021. Date of publication March 15, 2021; date of current version May 20, 2021. The work of Kai Wan and Giuseppe Caire was supported in part by the European Research Council under the ERC Advanced Grant 789190, CARENET. The work of Hua Sun was supported in part by the NSF under Award 2007108. The work of Mingyue Ji was supported in part by the NSF under Award 1817154 and Award 1824558. The work of Daniela Tuninetti was supported in part by the NSF under Award 1910309. This article was presented at the 2020 IEEE International Symposium on Information Theory. *(Corresponding author: Kai Wan.)*

Kai Wan and Giuseppe Caire are with the Electrical Engineering and Computer Science Department, Technische Universität Berlin, 10587 Berlin, Germany (e-mail: kai.wan@tu-berlin.de; caire@tu-berlin.de).

Hua Sun is with the Department of Electrical Engineering, University of North Texas, Denton, TX 76203 USA (e-mail: hua.sun@unt.edu).

Mingyue Ji is with the Electrical and Computer Engineering Department, The University of Utah, Salt Lake City, UT 84112 USA (e-mail: mingyue.ji@utah.edu).

Daniela Tuninetti is with the Electrical and Computer Engineering Department, University of Illinois at Chicago, Chicago, IL 60607 USA (e-mail: danielat@uic.edu).

Communicated by M. Kobayashi, Associate Editor for Communications.

Digital Object Identifier 10.1109/TIT.2021.3066005

I. Introduction

INFORMATION theoretic coded caching was originally proposed by Maddah-Ali and Niesen (MAN) [2] for the shared-link caching systems containing a server with a library of N equal-length files, which is connected to K users through a noiseless shared-link. Users can store up to M files in their local cache. Two phases are included in a caching system. Cache placement phase: content is pushed into each cache without knowledge of future demands. Delivery phase: each user demands one file and, according to the cache contents, the server broadcasts coded packets to all the users. The objective is to minimize the worst-case load, which is defined as the number of transmitted bits, normalized by the file length, needed to satisfy the all the users' demands regardless of the demands.

The MAN coded caching scheme proposed in [2] uses a combinatorial design in the placement phase (referred to as MAN uncoded cache placement), such that in the delivery phase binary multicast messages (referred to as MAN multicast messages) can simultaneously satisfy the users' demands. Under the constraint of uncoded cache placement (i.e., each user directly caches a subset of the library bits), the MAN scheme can achieve the minimum worst-case load among all possible demands when $\mathsf{N} \geq \mathsf{K}$ [3]. Noting that, if there are files demanded multiple times, some MAN multicast messages can be obtained as a binary linear combination of other MAN multicast messages, Yu, Maddah-Ali, and Avestimehr (YMA) proposed an improved delivery scheme that achieves the minimum worst-case load under the constraint of uncoded cache placement for $\mathsf{N} < \mathsf{K}$ [4]. The cost of uncoded cache placement compared to coded cache placement is known to be a multiplicative factor of at most two [5].

The MAN setting has been extended to numerous models, such as Device-to-Device (D2D) caching systems [6], coded caching with private demands [7]–[9], coded distributed computing [10], and coded data shuffling [11]–[13] – just to name a few. A common point of these models is that each user requests one file; some allow for users to request (the equivalent of) multiple files [10]–[15] which however does not substantially change the nature of the problem. In general, linear and multivariate polynomial operations are widely used in fundamental primitives for building the complex queries that support on-line analytics and data mining procedures. For example,

linear operations are critical in modern deep learning and artificial intelligence algorithms[1], while algebraic polynomial queries naturally arise in engineering problems such as those described by differential equations and distributed machine learning algorithms [16], [17]. In those scenarios, it may be too resource-consuming (in terms of bandwidth, or execution time, or storage space) to download locally all the input variables in order to compute the desired output value. Instead, it is desirable to directly download the result of the desired output function. This article studies the fundamental tradeoff between local storage and network load when users are interested in retrieving a function of the dataset available at the server. As the general function retrieval is a formidable difficult problem, this work considers the natural first generalization step: from single file retrieval to retrieving a general scalar linear function of the files.

The question we ask in this article is, compared to the original MAN caching problem, whether the optimal worst-case load is increased when the users are allowed to request *scalar linear functions of the files*. The original MAN caching problem in [2], where each user requests one file, is thus a special case of our cache-aided scalar linear function retrieval problem. In addition to the novel problem formulation, our main results are as follows.

1) *Achievable Cache-Aided Function Retrieval Scheme:* We propose a novel caching scheme to deliver scalar linear functions on an arbitrary finite field. To the best of our knowledge, even for the original MAN caching problem, no caching scheme is known to operate on an arbitrary field which achieves the same load as the YMA scheme. Compared to the YMA scheme, we use different encoding and decoding procedures for the multicast messages. On the observation that the YMA scheme fails on fields with characteristics strictly larger than two, we propose an encoding based on a finite field interference alignment idea and a decoding that needs both the sum and multiplication operations, while the YMA scheme is only defined on the binary field and thus uses only sum/XOR operations. Interestingly, the achieved load by the proposed scheme only depends on the number of linearly independent functions that are demanded, akin to the YMA's single-file retrieval scheme where the load depends on the number of distinct file requests. Surprisingly, we show that the load of our scheme is the same as the YMA load, when logarithms are taken with base equal to the size of the field.
   Finally, when the field is binary, the encoding procedure of our scheme reduces to the one of the YMA scheme. Following the decoding step of the YMA scheme, we provide an alternate and much simpler decodability proof for our proposed scheme compared to the one for the general case.
2) *Optimality:* On observation that the converse bound for the original MAN caching problem in [3], [4] is also a converse in the considered cache-aided function retrieval problem, we conclude that the proposed scheme achieves the optimal worst-case load under the constraint of uncoded cache placement and is order optimal to within a factor of two in general. From the results in this article, we can answer the question we asked at the beginning of this article: *the optimal worst-case load under the constraint of uncoded cache placement is not increased when the users are allowed to request scalar linear functions of the files.*
3) *Extensions:* A number of extensions of our problem formulation are possible.
   - As only associativity and commutativity are required to prove the correctness of our proposed decoding procedure, we conclude that the proposed caching scheme works not only on arbitrary finite field, but also on any commutative ring (i.e., a ring in which the multiplication operation is commutative).
   - Under the constraint of uncoded cache placement and that the subfile partition is symmetric across files (see Definition 1), we can show that the proposed caching scheme achieves the minimum average load over i.i.d. uniform demand distribution and over all possible demanded functions.
   - We then show how to extend the key idea of this article to scenarios including demand-private retrieval or device-to-device networks.
   - Finally, we note that any low subpacketization level scheme for the MAN problem works for scalar linear function retrieval as well if it has the following characteristics: (i) uses uncoded placement and whose subfile partition is symmetric across files (see Definition 1) and (ii) treats the demand of each user as an independent file. For example, this class of schemes includes schemes based on a grouping strategy, such as [18]–[20], schemes based on Placement Delivery Array (PDA), such as [21]–[23], and some other combinatorial designs, such as [24]–[27].

A. Paper Organization

The rest of this article is organized as follows. Section II formulates the cache-aided scalar function retrieval problem and introduces some related results in the literature. Section III discusses the main results in this article. Section IV describes the proposed achievable caching scheme. Section V concludes this article. Some of the proofs can be found in the Appendices.

[1]One of the simplest examples is as follows. Assume that one node k aims to perform a linear regression learning task. In this case, the goal of node k is to minimize $\frac{1}{2}\|\mathbf{A}\mathbf{x}_k - \mathbf{y}_k\|_2^2$ over all $\mathbf{x}_k \in \mathbb{R}$, where the matrix $\mathbf{A}$ is the entire dataset and is distributed over the nodes, and $\mathbf{y}_k$ is the observation. Then, at the time slot $t+1$, node k will compute $\mathbf{x}_k^{t+1} = \mathbf{x}_k^t - \eta\mathbf{A}^T(\mathbf{A}\mathbf{x}_k^t - \mathbf{y}_k)$ where η is the learning rate. In order to perform this computation task, it can be done into two steps. The first step is to compute $\mathbf{z}_k^t = \mathbf{A}\mathbf{x}_k^t - \mathbf{y}_k$ and the second step is to compute $\mathbf{x}_k^{t+1} = \mathbf{x}_k^t - \eta\mathbf{A}^T\mathbf{z}_k^t$. It can be seen that both steps involve different linear combinations of the entire data set and can be computed using the proposed approach. In addition, assume that another node j aims to compute the dominate eigenvector of the dataset $\mathbf{A}$. One popular approach is to use the power iteration algorithm. In this algorithm, at time slot t, it needs to compute $\mathbf{v}_j^{t+1} = \mathbf{A}\mathbf{v}_j^t/\|\mathbf{A}\mathbf{v}_j^t\|$, where $\mathbf{v}_j^t$ is the estimate of the dominant eigenvector of $\mathbf{A}$ in time slot t. It can be seen that this computation task at each time slot is also a linear operation, which can be performed using the proposed scheme.

B. Notation Convention

Calligraphic symbols denote sets, bold symbols denote vectors, and sans-serif symbols denote system parameters. We use $|\cdot|$ to represent the cardinality of a set or the length of a vector; $[a:b] := \{a, a+1, \ldots, b\}$ and $[n] := [1:n]$; $\oplus$ represents bit-wise XOR; $\mathbb{E}[\cdot]$ represents the expectation value of a random variable; $a! = a \times (a-1) \times \ldots \times 1$ represents the factorial of a; $\mathbb{F}_q$ represents a finite field with order q, where $q = p^n$ is a prime-power and the prime p is called the *characteristic* of this field; $\text{rank}_q(\mathbb{A})$ represents the rank of matrix $\mathbb{A}$ on field $\mathbb{F}_q$; $\det(\mathbb{A})$ represents the determinant of matrix $\mathbb{A}$; $\mathbb{A}_{\mathcal{S},\mathcal{V}}$ represents the sub-matrix of $\mathbb{A}$ by selecting from $\mathbb{A}$, the rows with indices in $\mathcal{S}$ and the columns with indices in $\mathcal{V}$. We let $\binom{x}{y} = 0$ if $x < 0$ or $y < 0$ or $x < y$. In this article, for each set of integers $\mathcal{S}$, we sort the elements in $\mathcal{S}$ in an increasing order and denote the ith smallest element by $\mathcal{S}(i)$, i.e., $\mathcal{S}(1) < \ldots < \mathcal{S}(|\mathcal{S}|)$.

II. System Model and Related Results

A. System Model

A $(\mathsf{K}, \mathsf{N}, \mathsf{q})$ shared-link cache-aided scalar linear function retrieval problem is defined as follows. A central server can access a library of N files. The files are denoted by $W_1, \ldots, W_{\mathsf{N}}$. Each file has F independent and uniformly distributed symbols over a finite field $\mathbb{F}_{\mathsf{q}}$, for some prime-power q. As in [2], we assume that F is sufficiently large such that any subpacketization of the files is possible. The central server is connected to K users through an error-free shared-link. Each user is equipped with a cache that can store up to MF symbols, where $\mathsf{M} \in [0, \mathsf{N}]$.

The system operates in two phases.

Cache Placement Phase. During the cache placement phase, each user stores information about the files in its local cache without knowledge of future users' demands, that is, there exist placement functions $\phi_k,\ k \in [\mathsf{K}]$, such that

$$\phi_k : [\mathbb{F}_{\mathsf{q}}]^{\mathsf{FN}} \to [\mathbb{F}_{\mathsf{q}}]^{\mathsf{FM}}, \tag{1}$$

where M denotes the *cache or memory size*. We denote the content in the cache of user $k \in [\mathsf{K}]$ by $Z_k = \phi_k(W_1, \ldots, W_{\mathsf{N}})$.

Delivery Phase. During the delivery phase, each user requests one scalar linear function of the files. The demand of user $k \in [\mathsf{K}]$ is represented by the row vector $\mathbf{y}_k = (y_{k,1}, \ldots, y_{k,\mathsf{N}}) \in [\mathbb{F}_{\mathsf{q}}]^{\mathsf{N}}$, which means that user k wants to retrieve the element-wise linear combination $y_{k,1}W_1 + \ldots + y_{k,\mathsf{N}}W_{\mathsf{N}}$. We denote the demand matrix of all users by

$$\mathbb{D} = [\mathbf{y}_1; \ldots; \mathbf{y}_{\mathsf{K}}] \in [\mathbb{F}_{\mathsf{q}}]^{\mathsf{K}\times\mathsf{N}}. \tag{2}$$

The demand matrix $\mathbb{D}$ is known to the server and all users. In addition, for each set $\mathcal{S} \subseteq [\mathsf{K}]$, we denote the demand matrix of the users in $\mathcal{S}$ by $\mathbb{D}_{\mathcal{S}} := \mathbb{D}_{\mathcal{S},[\mathsf{N}]}$.

Given the demand matrix $\mathbb{D}$, the server broadcasts the message $X = \psi(\mathbb{D}, W_1, \ldots, W_{\mathsf{N}})$ to each user $k \in [\mathsf{K}]$, where the encoding function ψ is such that

$$\psi : [\mathbb{F}_{\mathsf{q}}]^{\mathsf{K}\times\mathsf{N}} \times [\mathbb{F}_{\mathsf{q}}]^{\mathsf{FN}} \to [\mathbb{F}_{\mathsf{q}}]^{\mathsf{FR}}, \tag{3}$$

for some non-negative R referred to as *load*.

Decoding. Each user $k \in [\mathsf{K}]$ decodes its desired function from $(\mathbb{D}, Z_k, X)$. In other words, there exist decoding functions $\xi_k,\ k \in [\mathsf{K}]$, such that

$$\xi_k : [\mathbb{F}_{\mathsf{q}}]^{\mathsf{K}\times\mathsf{N}} \times [\mathbb{F}_{\mathsf{q}}]^{\mathsf{FM}} \times [\mathbb{F}_{\mathsf{q}}]^{\mathsf{FR}} \to [\mathbb{F}_{\mathsf{q}}]^{\mathsf{F}}, \tag{4}$$

$$\xi_k(\mathbb{D}, Z_k, X) = y_{k,1}W_1 + \ldots + y_{k,\mathsf{N}}W_{\mathsf{N}}. \tag{5}$$

Objective. For a given cache size $\mathsf{M} \in [0, \mathsf{N}]$, the objective is to determine the *minimum worst-case load* among all possible demands, defined as the smallest R such that there exist placement functions $\phi_k, k \in [\mathsf{K}]$, encoding function ψ, and decoding functions $\xi_k, k \in [\mathsf{K}]$, satisfying all the above constraints. The optimal load is denoted by $\mathsf{R}^\star$.

If each user directly copies some symbols of the files into its cache, the cache placement is said to be *uncoded*. The minimum worst-case load under the constraint of uncoded cache placement is denoted by $\mathsf{R}_{\mathsf{u}}^\star$.

B. Review of the MAN [2] and YMA [4] Schemes

In the following, we review the MAN and YMA coded caching schemes for the shared-link caching problem where each user requests one file. These schemes operate on the binary field $\mathbb{F}_2$.

1) MAN Scheme: **File Split.** Let $t \in [0:\mathsf{K}]$. Partition each file into $\binom{\mathsf{K}}{t}$ equal-length subfiles denoted by

$$W_i = \{W_{i,\mathcal{T}} : \mathcal{T} \subseteq [\mathsf{K}], |\mathcal{T}| = t\}, \quad i \in [\mathsf{N}]. \tag{6}$$

Placement Phase: User $k \in [\mathsf{K}]$ caches $W_{i,\mathcal{T}}, i \in [\mathsf{N}]$, if $k \in \mathcal{T}$. Hence, each user caches $\mathsf{N}\binom{\mathsf{K}-1}{t-1}$ subfiles, each of which contains $\frac{\mathsf{F}}{\binom{\mathsf{K}}{t}}$ symbols, which requires

$$\mathsf{M} = \mathsf{N}\binom{\mathsf{K}-1}{t-1} \Big/ \binom{\mathsf{K}}{t} = \mathsf{N}\frac{t}{\mathsf{K}}. \tag{7}$$

Delivery Phase: User $k \in [\mathsf{K}]$ requests the file with index $d_k \in [\mathsf{N}]$. The server then broadcasts the following *MAN multicast messages*: for each $\mathcal{S} \subseteq [\mathsf{K}]$ where $|\mathcal{S}| = t+1$, the server sends

$$X_{\mathcal{S}} = \bigoplus_{k\in\mathcal{S}} W_{d_k,\mathcal{S}\setminus\{k\}}. \tag{8}$$

The server thus sends $\binom{\mathsf{K}}{t+1}$ packets of $\frac{\mathsf{F}}{\binom{\mathsf{K}}{t}}$ symbols, which requires

$$\mathsf{R} = \binom{\mathsf{K}}{t+1} \Big/ \binom{\mathsf{K}}{t} = \frac{\mathsf{K}-t}{1+t}. \tag{9}$$

Decoding: The multicast message $X_{\mathcal{S}}$ in (8) is useful to each user $k \in \mathcal{S}$, since this user caches all subfiles contained in $X_{\mathcal{S}}$ except for the desired subfile $W_{d_k,\mathcal{S}\setminus\{k\}}$. Considering all multicast messages, each user can recover all uncached subfiles and thus recover its demanded file.

Load: The achieved memory-load tradeoff of the MAN scheme is the lower convex envelop of the following points

$$(\mathsf{M}, \mathsf{R}) = \left(\frac{\mathsf{N}t}{\mathsf{K}}, \frac{\mathsf{K}-t}{1+t}\right), \quad \forall t \in [0:\mathsf{K}]. \tag{10}$$

2) YMA Scheme: **File splitting** and **cache placement** are as for the MAN scheme.

Delivery Phase: The main idea of the YMA delivery is that, when a file is demanded by multiple users, some MAN multicast messages in (8) can be obtained as a linear combination of others. Thus the load of the MAN scheme in (9) can be further reduced by removing the redundant MAN multicast messages. More precisely, for each demanded file, randomly choose one user among all users demanding this file and designate it as the "leader user" for this file. Let $\mathcal{D} := \cup_{k\in[\mathsf{K}]}\{d_k\}$ be the set of all distinct files that are demanded, and $\mathcal{L}$ be the set of $|\mathcal{D}|$ leader users. The server only sends those multicast message $X_{\mathcal{S}}$ in (8) that are useful for the leader users, that is, if $\mathcal{S}\cap\mathcal{L}\neq\emptyset$, thus saving $\binom{\mathsf{K}-|\mathcal{D}|}{t+1}$ transmissions.

Decoding: Clearly, all leaders users can decode their demanded files as per the MAN scheme. The non-leader users appear to miss the multicast messages $X_{\mathcal{A}}$ for each $\mathcal{A}\subseteq[\mathsf{K}]$ where $\mathcal{A}\cap\mathcal{L}=\emptyset$ and $|\mathcal{A}|=t+1$. It was proved in [4, Lemma 1] that for any such fixed $\mathcal{A}$, we have

$$\bigoplus_{\mathcal{F}\in\mathscr{F}_{\mathcal{B}}} X_{\mathcal{B}\setminus\mathcal{F}}=0, \tag{11}$$

where $\mathcal{B}=\mathcal{A}\cup\mathcal{L}$. $\mathscr{F}_{\mathcal{B}}$ is the family of subsets $\mathcal{F}\subseteq\mathcal{B}$, where $|\mathcal{F}|=|\mathcal{D}|$ and each file in $\mathcal{D}$ is requested by exactly one user in $\mathcal{F}$. The key observation is that in $\bigoplus_{\mathcal{F}\in\mathscr{F}_{\mathcal{B}}} X_{\mathcal{B}\setminus\mathcal{F}}$ each involved subfile appears exactly twice (i.e., contained in two MAN multicast messages)[2], whose contribution on $\mathbb{F}_2$ is thus zero. From (11), we have

$$X_{\mathcal{A}}=\bigoplus_{\mathcal{F}\in\mathscr{F}_{\mathcal{B}}:\mathcal{F}\neq\mathcal{L}} X_{\mathcal{B}\setminus\mathcal{F}}, \tag{12}$$

thus the multicast message $X_{\mathcal{A}}$ can be recovered by all users from the delivery phase.

Load: The YMA scheme requires the load of

$$\mathsf{R}(\mathcal{D})=\frac{\binom{\mathsf{K}}{t+1}-\binom{\mathsf{K}-|\mathcal{D}|}{t+1}}{\binom{\mathsf{K}}{t}}, \tag{13}$$

if the set of the demanded files is $\mathcal{D}$. The worst-case load is attained for $|\mathcal{D}|=\min(\mathsf{N},\mathsf{K})$, thus the achieved memory-load tradeoff of the YMA scheme is the lower convex envelop of the following points

$$(\mathsf{M},\mathsf{R})=\left(\frac{\mathsf{N}t}{\mathsf{K}},\frac{\binom{\mathsf{K}}{t+1}-\binom{\mathsf{K}-\min(\mathsf{N},\mathsf{K})}{t+1}}{\binom{\mathsf{K}}{t}}\right),\quad \forall t\in[0:\mathsf{K}]. \tag{14}$$

III. Main Results and Discussion

In this section, we summarize the main results in this article and provide some discussion.

A. Achievability

The main result of this article is as follows.

Theorem 1 (Achievability): For the $(\mathsf{K},\mathsf{N},\mathsf{q})$ shared-link cache-aided scalar linear function retrieval problem, the YMA load in (14) is an achievable worst-case load. More precisely, for cache size $\mathsf{M}=\frac{\mathsf{N}t}{\mathsf{K}}$, with $t\in[0:\mathsf{K}]$, and for demand matrix $\mathbb{D}$, the load

$$\mathsf{R}(\mathbb{D}):=\frac{\binom{\mathsf{K}}{t+1}-\binom{\mathsf{K}-\mathrm{rank}_{\mathsf{q}}(\mathbb{D})}{t+1}}{\binom{\mathsf{K}}{t}} \tag{15}$$

is achievable. The worst-case load is attained by $\mathrm{rank}_{\mathsf{q}}(\mathbb{D})=\min(\mathsf{N},\mathsf{K})$. □

Remark 1 (Dependance on the Rank of the Demand Matrix): The load in (15) is a generalization of the load in (13) achieved by the YMA scheme. More precisely, if each user $k\in[\mathsf{K}]$ requests one file (i.e., $\mathbf{y}_k\in[0:1]^{\mathsf{N}}$ with a unit norm), $\mathrm{rank}_{\mathsf{q}}(\mathbb{D})$ is exactly the number of demanded files, and thus the proposed scheme achieves the load in (13) as the YMA scheme. Interestingly, the load of the proposed scheme only depends on the rank of the demand matrix instead of on the specifically demanded functions. □

Remark 2 (High-Level Ideas for Theorem 1 and Its Comparison to the YMA Scheme): We partition the "symbol positions" set $[\mathsf{F}]$ as follows

$$[\mathsf{F}]=\bigcup_{\mathcal{T}\subseteq[\mathsf{K}]:|\mathcal{T}|=t}\mathcal{I}_{\mathcal{T}} \text{ such that } |\mathcal{I}_{\mathcal{T}}|=\mathsf{F}/\binom{\mathsf{K}}{t}. \tag{16}$$

Then, with a Matlab-inspired notation, we let

$$W_{i,\mathcal{T}}=W_i(\mathcal{I}_{\mathcal{T}}),\ \forall\mathcal{T}\subseteq[\mathsf{K}]:|\mathcal{T}|=t,\quad\forall i\in[\mathsf{N}], \tag{17}$$

representing the ordered set of symbols of W_i whose position is in $\mathcal{I}_{\mathcal{T}}$. As in the MAN placement, user $k\in[\mathsf{K}]$ caches $W_{i,\mathcal{T}}$ if $k\in\mathcal{T}$ for all $i\in[\mathsf{N}]$. By doing so, any scalar linear function is naturally partitioned into "blocks" as follows

$$\begin{aligned}&y_{k,1}W_1+\ldots+y_{k,\mathsf{N}}W_{\mathsf{N}}\\&=\{\underbrace{y_{k,1}W_1(\mathcal{I}_{\mathcal{T}})+\ldots+y_{k,\mathsf{N}}W_{\mathsf{N}}(\mathcal{I}_{\mathcal{T}})}_{:=B_{k,\mathcal{T}}\text{ is the }\mathcal{T}\text{-th block of the }k\text{-th demanded function}}:\mathcal{T}\subseteq[\mathsf{K}],|\mathcal{T}|=t\}.\end{aligned} \tag{18}$$

Some blocks of the demanded functions can thus be computed based on the cache content available at each user while the remaining ones need to be delivered by the server. With this specific file split (and corresponding MAN cache placement), *we operate the MAN delivery scheme over the blocks instead of over the subfiles*; more precisely, instead of (8) we transmit

$$X_{\mathcal{S}}=\sum_{k\in\mathcal{S}}\alpha_{\mathcal{S},k}B_{k,\mathcal{S}\setminus\{k\}},\forall\mathcal{S}\subseteq[\mathsf{K}]:|\mathcal{S}|=t+1, \tag{19}$$

for some non-zero encoding coefficient $\alpha_{\mathcal{S},k}\in\mathbb{F}_{\mathsf{q}}$ in (19).

Clearly, this scheme achieves the same load as in (9) and works on any finite field.

The question is whether we can do something similar to the YMA scheme with (19). More specifically, we seek answers to the following questions.

1) What is a suitable definition of the leader user set $\mathcal{L}$?
2) What is a suitable choice of the encoding coefficients $\alpha_{\mathcal{S},k}$ for all $\mathcal{S}\subseteq[\mathsf{K}]$, $|\mathcal{S}|=t+1$, and $k\in\mathcal{S}$ in (19)?
3) Assuming we only send the multicast messages that are useful for the leaders (i.e., $X_{\mathcal{S}}$ where $\mathcal{S}\subseteq[\mathsf{K}]$, $|\mathcal{S}|=t+1$, and $\mathcal{S}\cap\mathcal{L}\neq\emptyset$), what is the counterpart of (12)?

[2] In this article, A 'appears' in a linear combination means that in the linear combination, there exists some term in the linear combination including A. A linear combination 'contains' B means that in the linear combination, the coefficient multiplying B is not 0. For example, we say A appears in the linear combination $(A\oplus B)\oplus(A\oplus C)$, but the linear combination does not contain A.

Here, for each $\mathcal{A} \subseteq [\mathsf{K}]$ where $|\mathcal{A}| = t+1$ and $\mathcal{A} \cap \mathcal{L} = \emptyset$, we seek

$$X_{\mathcal{A}} = \sum_{\mathcal{S} \subseteq [\mathsf{K}]: |\mathcal{S}| = t+1, \mathcal{S} \cap \mathcal{L} \neq \emptyset} \beta_{\mathcal{A},\mathcal{S}} X_{\mathcal{S}}, \tag{20}$$

for some decoding coefficient $\beta_{\mathcal{A},\mathcal{S}} \in \mathbb{F}_{\mathsf{q}}$ in (20).

The novelty of our scheme lays in the answers to these questions as follows.

1) We first choose $\text{rank}_{\mathsf{q}}(\mathbb{D})$ leaders whose demand vectors are linearly independent, and denote the set of leader users by $\mathcal{L}$. By construction, the demand sub-matrix of the leader users is full-rank.
2) One can see that if the encoding coefficients in (19) are taken be equal to 1, as in the YMA scheme, and only the multicast messages useful for at least one leader are sent, as in the YMA scheme, then it is not possible to reconstruct the multicast messages useful for non-leaders only whenever the characteristic of the field is strictly larger than two–see Remark 4.
   Instead, for any prime-power q, we propose to separate the blocks demanded by the leaders from those demanded by non-leaders in (19) as
   $$X_{\mathcal{S}} = \sum_{k_1 \in \mathcal{S} \cap \mathcal{L}} \alpha_{\mathcal{S},k_1} B_{k_1,\mathcal{S}\setminus\{k_1\}} + \sum_{k_2 \in \mathcal{S} \setminus \mathcal{L}} \alpha_{\mathcal{S},k_2} B_{k_2,\mathcal{S}\setminus\{k_2\}}. \tag{21}$$
   We then propose to alternate the encoding coefficients of the desired blocks by the leaders (i.e., users in $\mathcal{S} \cap \mathcal{L}$) between $+1$ and -1, i.e., the encoding coefficient of the desired block of the first leader is $+1$, the encoding coefficient of the desired block of the second leader is -1, the encoding coefficient of the desired block of the third leader is $+1$, etc. Similarly, we propose to alternate the encoding coefficients of the desired blocks by the non-leaders (i.e., users in $\mathcal{S} \setminus \mathcal{L}$) between $+1$ and -1.[3]
3) With the above encoding coefficients, we compute the decoding coefficients by equating (21) to (20) for every $\mathcal{A} \subseteq [\mathsf{K}]$ where $|\mathcal{A}| = t+1$ and $\mathcal{A} \cap \mathcal{L} = \emptyset$; the solution is given in (57b). With this, each user can recover all multicast messages, sent or not, and thus it can recover its desired function.
4) For $\mathsf{q} = 2$ we have $-1 = 1$ (i.e., the choice of the encoding coefficients is exactly the same as the YMA scheme). In this case, following the decoding step of the YMA scheme that only uses sum/XOR operations (reviewed in Section II-B), we propose an alternate and much simpler decodability proof than the one for the general q. Note that for the case $\mathsf{q} = 2^n$ where $n > 1$, even if the choice of the encoding coefficients is exactly the same as the YMA scheme[4], the YMA scheme fails as in this case multiplication operations are needed during the decoding step–See Remark 5. □

[3] This type of code was originally proposed for the private function retrieval problem [28], where there is a memory-less user aiming to retrieve a scalar linear function of the files in the library from multiple servers (each server can access the whole library), while preserving the demand of this user from each server.

[4] If the characteristic of a field is 2, for each element on this field (denoted by a), we have $a + a = 0$. Hence, $+1$ and -1 are both equal to 1.

B. Optimality

As the setting where each user demands one file is a special case of the considered cache-aided scalar linear function retrieval problem, each of the converse bounds in [3]–[5] for the original shared-link coded caching problem is also a converse in our considered problem, thus we immediately have the following optimality guarantees.

Theorem 2 (Optimality): For the $(\mathsf{K}, \mathsf{N}, \mathsf{q})$ shared-link cache-aided scalar linear function retrieval problem, under the constraint of uncoded cache placement, the optimal worst-case load-memory tradeoff is the lower convex envelop of

$$(\mathsf{M}, \mathsf{R}_{\mathsf{u}}^{\star}) = \left(\frac{\mathsf{N}t}{\mathsf{K}}, \frac{\binom{\mathsf{K}}{t+1} - \binom{\mathsf{K}-\min\{\mathsf{K},\mathsf{N}\}}{t+1}}{\binom{\mathsf{K}}{t}} \right), \ \forall t \in [0:\mathsf{K}]. \tag{22}$$

Moreover, the achieved worst-case load in (22) is optimal to within a factor of two in general.

□

C. Extensions

We discuss here the extensions of the proposed caching scheme in Theorem 1.

1) Extension to Commutative Rings and Monomial Retrieval: As only associativity and commutativity are required to prove the correctness of the decoding procedure for the scheme in Theorem 1, we conclude that our scheme works on any commutative ring as well. With this observation, we can further extend the scope of this work so as to include monomial retrieval in the spirit of [29]. The idea is that one can define a discrete logarithm that maps each non-zero element on the finite field $\mathbb{F}_{\mathsf{q}}$ to a distinct element on the integral domain $\mathbb{Z}_{\mathsf{q}-1}$. Thus, except for a special treatment that one has to reserve to the case where one of the factors in the monomial is zero, we can use our proposed scheme on the integral domain $\mathbb{Z}_{\mathsf{q}-1}$.

2) Optimal Average Load Under Uncoded and Symmetric Cache Placement: We define *uncoded and symmetric* cache placement as follows, as a generalization of (16)-(17).

Definition 1 (Uncoded and Symmetric Cache Placement): We partition the symbol positions set into disjoint index sets,

$$[\mathsf{F}] = \bigcup_{\mathcal{T} \subseteq [\mathsf{K}]} \mathcal{I}_{\mathcal{T}}, \tag{23}$$

where $\mathcal{I}_{\mathcal{T}}$ can be empty for some $\mathcal{T} \subseteq [\mathsf{K}]$. We then let $W_{i,\mathcal{T}} = W_i(\mathcal{I}_{\mathcal{T}})$ for all $\mathcal{T} \subseteq [\mathsf{K}]$ and $i \in [\mathsf{N}]$ as in (17). Each user $k \in [\mathsf{K}]$ caches $W_{i,\mathcal{T}}$ if $k \in \mathcal{T}$ for all $i \in [\mathsf{N}]$. □

In the delivery phase, user $k \in [\mathsf{K}]$ needs to recover $B_{k,\mathcal{T}}$ (defined in (18)) for all $\mathcal{T} \subseteq [\mathsf{K}] \setminus \{k\}$. By directly using [5, Lemma 2] in the caching converse bound under uncoded cache placement in [3], [4], we can prove that the proposed caching scheme in Theorem 1 achieves the optimal *average load* over i.i.d. uniform demand distribution under the constraint of uncoded and symmetric cache placement.

Corollary 1 (Optimal Average Load Under Uncoded and Symmetric Cache Placement): For the $(\mathsf{K},\mathsf{N},\mathsf{q})$ shared-link cache-aided scalar linear function retrieval problem, under the constraint of uncoded and symmetric cache placement, the optimal average load over i.i.d. uniform demand distribution is the lower convex envelop of

$$(\mathsf{M},\mathsf{R}) = \left(\frac{\mathsf{N}t}{\mathsf{K}}, \mathbb{E}_{\mathbb{D}}\left[\frac{\binom{\mathsf{K}}{t+1} - \binom{\mathsf{K}-\mathrm{rank}_{\mathsf{q}}(\mathbb{D})}{t+1}}{\binom{\mathsf{K}}{t}}\right]\right),\ \forall t \in [0:\mathsf{K}]. \tag{24}$$

Note that an uncoded and asymmetric cache placement can be treated as a special case of the inter-file coded cache placement in the original MAN caching problem. It is part of on-going work to derive a converse bound under the constraint of uncoded cache placement (i.e., without constraint of symmetric cache placement) for the considered cache-aided function retrieval problem.

3) Device-to-Device (D2D) Cache-Aided Scalar Linear Function Retrieval: Coded caching was originally used in Device-to-Device networks in [6], where in the delivery phase each user broadcasts packets, which are functions of its cached content and the users' demands, to all other users. The authors in [30] extended the YMA scheme to D2D networks by dividing the D2D networks into K shared-link networks, and used the YMA scheme in each shared-link network. Hence, when users request scalar linear functions, we can use the method in [30] to divide the D2D networks into K shared-link networks, and then use the proposed caching scheme in Theorem 1 in each shared-link network.

Corollary 2 (D2D Cache-Aided Scalar Linear Function Retrieval): For the $(\mathsf{K},\mathsf{N},\mathsf{q})$ D2D cache-aided scalar linear function retrieval problem, the lower convex envelop of the following points is achievable

$$(\mathsf{M},\mathsf{R}) = \left(\frac{\mathsf{N}t}{\mathsf{K}}, \max_{\mathbb{D}} \frac{\binom{\mathsf{K}-1}{t} - \frac{1}{\mathsf{K}}\sum_{k\in[\mathsf{K}]}\binom{\mathsf{K}-1-\mathrm{rank}_{\mathsf{q}}(\mathbb{D}_{[\mathsf{K}]\setminus\{k\}})}{t}}{\binom{\mathsf{K}-1}{t-1}}\right), \tag{25}$$

for all $t \in [\mathsf{K}]$.

4) Cache-Aided Private Scalar Linear Function Retrieval: For successful decoding in the proposed scheme in Theorem 1, users need to be aware of the demands of other users, which is not private. To preserve the privacy of the demand of a user against the other users, one can design schemes that include virtual users as in [7], [8]. In the proposed scheme in [8], each possible demanded function (the total number of possible demanded functions is $\mathsf{N}' := \frac{\mathsf{q}^{\mathsf{N}}-1}{\mathsf{q}-1}$) is demanded exactly K times. Thus there are totally $\mathsf{N}'\mathsf{K}$ real or virtual users in the system. Then the proposed scheme in Theorem 1 can be used to satisfy the demands of all real or virtual users. As each user cannot distinguish other real users from virtual users, the resulting scheme does not leak any information on the demands of real users.

Corollary 3 (Cache-Aided Private Scalar Linear Function Retrieval): For the $(\mathsf{K},\mathsf{N},\mathsf{q})$ shared-link cache-aided private scalar linear function retrieval problem, the lower convex envelop of the following points is achievable

$$(\mathsf{M},\mathsf{R}) = \left(\frac{t}{\mathsf{N}'\mathsf{K}}\mathsf{N}, \frac{\binom{\mathsf{N}'\mathsf{K}}{t+1} - \binom{\mathsf{N}'\mathsf{K}-\mathsf{N}}{t+1}}{\binom{\mathsf{N}'\mathsf{K}}{t}}\right),\quad \forall t \in [\mathsf{N}'\mathsf{K}]. \tag{26}$$

Remark 3: Very recently, the authors in [31] found an important application of our proposed cache-aided linear function retrieval scheme in the setting where the privacy of each user's demand against the other users must be preserved. The key idea in [31] is as follows. In the placement phase, besides the content which users should cache in the MAN cache placement, users also cache some random linear combinations of the files that are not cached in the MAN scheme, which act as a privacy key. In the delivery phase, the server delivers multicast messages as if the demands are linear combinations of the files, where the linear combination for a user is determined by its privacy key and its demand. Compared to existing private caching schemes based on virtual users, the scheme in [31] achieves a similar load while reducing the subpacketization level from $2^{\mathsf{NK}\mathcal{H}(\mathsf{M}/\mathsf{N})}$ to $2^{\mathsf{K}\mathcal{H}(\mathsf{M}/\mathsf{N})}$, where $\mathcal{H}(p) = -p\log_2(p) - (1-p)\log_2(1-p)$ is the binary entropy function. This reduction is an exponential improvement on the subpacketization level. □

5) Reduction on the Subpacketization Level: In practical scenarios, the high subpacketization level of any scheme based on the MAN placement, which increases exponentially with the number of users K, is problematic. Low subpacketization level schemes for the MAN problem have been extensively studied in the literature. Here we remark that any low subpacketization level scheme for the MAN problem, which (i) uses uncoded and symmetric cache placement and (ii) treats the demand of each user as an independent file, works for scalar linear function retrieval as well – for example, schemes based on grouping strategy, such as [18]–[20], schemes based on Placement Delivery Array (PDA), such as [21]–[23], and some other combinatorial designs such as [24]–[27]. Similarly to the scheme achieving the MAN load introduced in this article, the solution for the delivery phase is to treat each demanded linear function as an independent file and partition each linear function into blocks, which are treated as subfiles in the original problem.

IV. Achievable Scheme in Theorem 1 for General Prime-Power q

In the following, we describe the proposed cache-aided function retrieval scheme in Theorem 1 where the users' demands are scalar linear functions on arbitrary finite field. All the operations in the proposed scheme are on the same finite field. First, we give an example to illustrate the main ideas of the proposed scheme. Next, we present the general cache-aided function retrieval scheme for any prime-power q. Finally, for the case $\mathsf{q} = 2$, where the proposed scheme reduces to the YMA scheme, we give an alternate decodability proof that is simpler than the one for the general case.

A. Example $\mathsf{K} = 4, \mathsf{N} = 2, t = 1$

Consider the $(\mathsf{K},\mathsf{N},\mathsf{q}) = (4,2,\mathsf{q})$ shared-link cache-aided scalar linear function retrieval problem, where q is a prime-power. We consider $t = \mathsf{KM}/\mathsf{N} = 1$, that is, $\mathsf{M} = 1/2$. Each

file is partitioned into $\binom{\mathsf{K}}{t} = 4$ equal-length subfiles. We use the file split in (16)-(17), resulting in the demand split in (18).

In the delivery phase, we assume that

$$\begin{aligned}&\text{user 1 demands } W_1;\\&\text{user 2 demands } W_2;\\&\text{user 4 demands } y_{3,1}\ W_1 + y_{3,2}\ W_2;\\&\text{user 5 demands } y_{4,1}\ W_1 + y_{4,2}\ W_2;\end{aligned}$$

i.e., the demand matrix is

$$\mathbb{D} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ y_{3,1} & y_{3,2} \\ y_{4,1} & y_{4,2} \end{bmatrix} \in [\mathbb{F}_{\mathsf{q}}]^{4\times 2}. \tag{27}$$

We choose the set of leaders $\mathcal{L} = [2]$, since $\text{rank}_{\mathsf{q}}(\mathbb{D}_{[2]}) = 2$. Each user $k \in [4]$ should recover each block $B_{k,\mathcal{T}} = y_{k,1}\ W_{1,\mathcal{T}} + y_{k,2}\ W_{2,\mathcal{T}}$ after the delivery phase, where $\mathcal{T} \in [4] \setminus \{k\}$ and $|\mathcal{T}| = 1$. Note that $y_{1,1} = y_{2,2} = 1$ and $y_{1,2} = y_{2,1} = 0$.

Encoding. For each set $\mathcal{S} \subseteq [\mathsf{K}]$ where $|\mathcal{S}| = t+1 = 2$, the multicast messages in (19) are

$$X_{\{1,2\}} = \alpha_{\{1,2\},1}\ B_{1,\{2\}} + \alpha_{\{1,2\},2}\ B_{2,\{1\}}, \quad \text{(all leaders);} \tag{28a}$$

$$X_{\{1,3\}} = \alpha_{\{1,3\},1}\ B_{1,\{3\}} + \alpha_{\{1,3\},3}\ B_{3,\{1\}}, \quad \text{(mixed);} \tag{28b}$$

$$X_{\{1,4\}} = \alpha_{\{1,4\},1}\ B_{1,\{4\}} + \alpha_{\{1,4\},4}\ B_{4,\{1\}}, \quad \text{(mixed);} \tag{28c}$$

$$X_{\{2,3\}} = \alpha_{\{2,3\},2}\ B_{2,\{3\}} + \alpha_{\{2,3\},3}\ B_{3,\{2\}}, \quad \text{(mixed);} \tag{28d}$$

$$X_{\{2,4\}} = \alpha_{\{2,4\},2}\ B_{2,\{4\}} + \alpha_{\{2,4\},4}\ B_{4,\{2\}}, \quad \text{(mixed);} \tag{28e}$$

$$X_{\{3,4\}} = \alpha_{\{3,4\},3}\ B_{3,\{4\}} + \alpha_{\{3,4\},4}\ B_{4,\{3\}}, \quad \text{(all non-leaders).} \tag{28f}$$

Hence, there are $(t+1)\binom{\mathsf{K}}{t+1} = 12$ non-zero encoding coefficients in (28). Our objective is to express $X_{\{3,4\}}$ as a linear combination of the other multicast messages, so that we only need to transmit $\binom{\mathsf{K}}{t+1} - \binom{\mathsf{K}-|\mathcal{L}|}{t+1} = 5$ multicast messages. In other words, we aim to find 5 decoding coefficients such that

$$\begin{aligned} X_{\{3,4\}} &= \beta_{\{3,4\},\{1,2\}}\ X_{\{1,2\}} + \beta_{\{3,4\},\{1,3\}}\ X_{\{1,3\}} \\ &\quad + \beta_{\{3,4\},\{1,4\}}\ X_{\{1,4\}} + \beta_{\{3,4\},\{2,3\}}\ X_{\{2,3\}} \\ &\quad + \beta_{\{3,4\},\{2,4\}}\ X_{\{2,4\}} \qquad (29a)\\ &= \beta_{\{1,2\}}\ X_{\{1,2\}} + \beta_{\{1,3\}}\ X_{\{1,3\}} + \beta_{\{1,4\}}\ X_{\{1,4\}} \\ &\quad + \beta_{\{2,3\}}\ X_{\{2,3\}} + \beta_{\{2,4\}}\ X_{\{2,4\}}. \qquad (29b)\end{aligned}$$

In this example, only $X_{\{3,4\}}$ must be reconstructed from the other multicast messages, thus we simply the notation in the rest of this example and use $\beta_{\mathcal{S}}$ rather than $\beta_{\mathcal{A},\mathcal{S}}$.

We seek a solution for

$$\begin{aligned} &\alpha_{\{3,4\},3}\ (y_{3,1}\ W_{1,\{4\}} + y_{3,2}\ W_{2,\{4\}}) \\ &+ \alpha_{\{3,4\},4}\ (y_{4,1}\ W_{1,\{3\}} + y_{4,2}\ W_{2,\{3\}}) \qquad (30a)\\ &= \beta_{\{1,2\}}\ \big(\alpha_{\{1,2\},1}\ W_{1,\{2\}} + \alpha_{\{1,2\},2}\ W_{2,\{1\}}\big) \\ &+ \beta_{\{1,3\}}\ \big(\alpha_{\{1,3\},1}W_{1,\{3\}} + \alpha_{\{1,3\},3}(y_{3,1}W_{1,\{1\}} + y_{3,2}W_{2,\{1\}})\big) \\ &+ \beta_{\{1,4\}}\ \big(\alpha_{\{1,4\},1}W_{1,\{4\}} + \alpha_{\{1,4\},4}(y_{4,1}W_{1,\{1\}} + y_{4,2}W_{2,\{1\}})\big) \\ &+ \beta_{\{2,3\}}\ \big(\alpha_{\{2,3\},2}W_{2,\{3\}} + \alpha_{\{2,3\},3}(y_{3,1}W_{1,\{2\}} + y_{3,2}W_{2,\{2\}})\big) \\ &+ \beta_{\{2,4\}}\big(\alpha_{\{2,4\},2}W_{2,\{4\}} + \alpha_{\{2,4\},4}(y_{4,1}W_{1,\{2\}} + y_{4,2}W_{2,\{2\}})\big). \qquad (30b)\end{aligned}$$

We proceed to equate the coefficients on the left and on the right hand side of (30). We start with the easy equations, namely

$$\begin{aligned} &\text{for } W_{1,\{3\}} : y_{4,1}\ \alpha_{\{3,4\},4} = \beta_{\{1,3\}}\ \alpha_{\{1,3\},1} \\ &\iff \beta_{\{1,3\}} = y_{4,1}\ \frac{\alpha_{\{3,4\},4}}{\alpha_{\{1,3\},1}}, \qquad (31a)\\ &\text{for } W_{1,\{4\}} : y_{3,1}\ \alpha_{\{3,4\},3} = \beta_{\{1,4\}}\ \alpha_{\{1,4\},1} \\ &\iff \beta_{\{1,4\}} = y_{3,1}\ \frac{\alpha_{\{3,4\},3}}{\alpha_{\{1,4\},1}}, \qquad (31b)\\ &\text{for } W_{2,\{3\}} : y_{4,2}\ \alpha_{\{3,4\},4} = \beta_{\{2,3\}}\ \alpha_{\{2,3\},2} \\ &\iff \beta_{\{2,3\}} = y_{4,2}\ \frac{\alpha_{\{3,4\},4}}{\alpha_{\{2,3\},2}}, \qquad (31c)\\ &\text{for } W_{2,\{4\}} : y_{3,2}\ \alpha_{\{3,4\},3} = \beta_{\{2,4\}}\ \alpha_{\{2,4\},2} \\ &\iff \beta_{\{2,4\}} = y_{3,2}\ \frac{\alpha_{\{3,4\},3}}{\alpha_{\{2,4\},2}}. \qquad (31d)\end{aligned}$$

Then we move to the following equations

$$\begin{aligned} &\text{for } W_{1,\{1\}} : 0 = y_{3,1}\ \beta_{\{1,3\}}\ \alpha_{\{1,3\},3} + y_{4,1}\ \beta_{\{1,4\}}\ \alpha_{\{1,4\},4} \\ &\overset{(31a)(31b)}{\iff} \frac{\alpha_{\{3,4\},4}\ \alpha_{\{1,3\},3}}{\alpha_{\{1,3\},1}} + \frac{\alpha_{\{3,4\},3}\ \alpha_{\{1,4\},4}}{\alpha_{\{1,4\},1}} = 0, \qquad (32a)\end{aligned}$$

$$\begin{aligned} &\text{for } W_{2,\{2\}} : 0 = y_{3,2}\ \beta_{\{2,3\}}\ \alpha_{\{2,3\},3} + y_{4,2}\ \beta_{\{2,4\}}\ \alpha_{\{2,4\},4} \\ &\overset{(31c)(31d)}{\iff} \frac{\alpha_{\{3,4\},4}\ \alpha_{\{2,3\},3}}{\alpha_{\{2,3\},2}} + \frac{\alpha_{\{3,4\},3}\ \alpha_{\{2,4\},4}}{\alpha_{\{2,4\},2}} = 0, \qquad (32b)\end{aligned}$$

which imply the following alignment condition (note the minus sign!)

$$-\frac{\alpha_{\{3,4\},4}}{\alpha_{\{3,4\},3}} = \frac{\alpha_{\{1,4\},4}}{\alpha_{\{1,4\},1}}\ \frac{\alpha_{\{1,3\},1}}{\alpha_{\{1,3\},3}} = \frac{\alpha_{\{2,4\},4}}{\alpha_{\{2,4\},2}}\ \frac{\alpha_{\{2,3\},2}}{\alpha_{\{2,3\},3}}. \tag{33}$$

Finally, with the condition in (33) and the previously determined decoding coefficients, we get

$$\begin{aligned} &\text{for } W_{2,\{1\}} : \\ &-\beta_{\{1,2\}}\ \alpha_{\{1,2\},2} = y_{3,2}\ \beta_{\{1,3\}}\ \alpha_{\{1,3\},3} + y_{4,2}\ \beta_{\{1,4\}}\ \alpha_{\{1,4\},4} \\ &= (-y_{4,1}\ y_{3,2} + y_{3,1}\ y_{4,2})\ \frac{\alpha_{\{3,4\},3}\ \alpha_{\{1,4\},4}}{\alpha_{\{1,4\},1}}, \qquad (34a)\\ &\text{for } W_{1,\{2\}} : \\ &-\beta_{\{1,2\}}\ \alpha_{\{1,2\},1} = y_{3,1}\ \beta_{\{2,3\}}\ \alpha_{\{2,3\},3} + y_{4,1}\ \beta_{\{2,4\}}\ \alpha_{\{2,4\},4} \\ &= (-y_{4,1}\ y_{3,2} + y_{3,1}\ y_{4,2})\ \frac{\alpha_{\{3,4\},4}\ \alpha_{\{2,3\},3}}{\alpha_{\{2,3\},2}}, \qquad (34b)\end{aligned}$$

which imply another alignment condition

$$\frac{\alpha_{\{3,4\},4}}{\alpha_{\{3,4\},3}} = \frac{\alpha_{\{2,3\},2}\ \alpha_{\{1,4\},4}\ \alpha_{\{1,2\},1}}{\alpha_{\{2,3\},3}\ \alpha_{\{1,4\},1}\ \alpha_{\{1,2\},2}}. \tag{35}$$

To satisfy the constraints in (31)-(35), we choose the following solution. In each $X_{\mathcal{S}}$ where $\mathcal{S} \subseteq [4]$ and $|\mathcal{S}| = 2$,

we first alternate the encoding coefficients (either 1 or -1) of the desired blocks of the leaders in $\mathcal{S}$, and then alternate the encoding coefficients (either 1 or -1) of the desired blocks of the non-leaders in $\mathcal{S}$. With this, the multicast messages are

$$X_{\{1,2\}} = B_{1,\{2\}} - B_{2,\{1\}}; \tag{36a}$$
$$X_{\{1,3\}} = B_{1,\{3\}} + B_{3,\{1\}}; \tag{36b}$$
$$X_{\{1,4\}} = B_{1,\{4\}} + B_{4,\{1\}}; \tag{36c}$$
$$X_{\{2,3\}} = B_{2,\{3\}} + B_{3,\{2\}}; \tag{36d}$$
$$X_{\{2,4\}} = B_{2,\{4\}} + B_{4,\{2\}}; \tag{36e}$$
$$X_{\{3,4\}} = B_{3,\{4\}} - B_{4,\{3\}}. \tag{36f}$$

With the above choice on the encoding coefficients, we can check that all alignment conditions are satisfied with the decoding coefficients,

$$\beta_{\{1,2\}} = y_{3,1}\ y_{4,2} - y_{4,1}\ y_{3,2} = \det([y_{3,1}, y_{3,2}; y_{4,1}, y_{4,2}]); \tag{37a}$$
$$\beta_{\{1,3\}} = -y_{4,1} = -\det([y_{4,1}]); \tag{37b}$$
$$\beta_{\{1,4\}} = y_{3,1} = \det([y_{3,1}]); \tag{37c}$$
$$\beta_{\{2,3\}} = -y_{4,2} = -\det([y_{4,2}]); \tag{37d}$$
$$\beta_{\{2,4\}} = y_{3,2} = \det([y_{3,2}]). \tag{37e}$$

Delivery. The server broadcasts $X_{\mathcal{S}}$ for each $\mathcal{S} \subseteq [4]$ where $|\mathcal{S}| = 2$ and $\mathcal{S} \cap [2] \neq \emptyset$. In other words, the server broadcasts all the multicast messages in (36) except for $X_{\{3,4\}}$.

Decoding. As shown in (29b), each user can recover all multicast messages in (36), such that it can then recover its demanded function.

Performance. In total we transmit 5 multicast messages, each of which contains $\frac{1}{4}$ of the file symbols. Hence, the transmitted load is $\frac{5}{4}$, which coincides with the optimal worst-case load in Theorem 2.

Remark 4: In this example, the key novelty of the proposed scheme is to determine the encoding coefficients. The solution is that in each multicast message we alternate the signs (either 1 or -1) of the encoding coefficients for the desired blocks by the leaders and by the non-leaders, respectively. However, in the YMA scheme all the encoding coefficients are set to 1. In the following, we show that for an arbitrary field $\mathbb{F}_{\mathsf{q}}$ of characteristic larger than 2, if we choose the encoding coefficients as the YMA scheme, the resulting scheme is not decodable.

In the above example, let $[y_{3,1}, y_{3,2}] = [1, 1]$ and $[y_{4,1}, y_{4,2}] = [1, 0]$. In the delivery phase, with the set of leaders $\mathcal{L} = [2]$, we generate the multicast messages as for the YMA scheme

$$X_{\{1,2\}} = W_{1,\{2\}} + W_{2,\{1\}}; \tag{38a}$$
$$X_{\{1,3\}} = W_{1,\{3\}} + (W_{1,\{1\}} + W_{2,\{1\}}); \tag{38b}$$
$$X_{\{1,4\}} = W_{1,\{4\}} + W_{1,\{1\}}; \tag{38c}$$
$$X_{\{2,3\}} = W_{2,\{3\}} + (W_{1,\{2\}} + W_{2,\{2\}}); \tag{38d}$$
$$X_{\{2,4\}} = W_{2,\{4\}} + W_{1,\{2\}}. \tag{38e}$$

Let us then focus on user 3 who desires $W_1 + W_2$. From its cache, user 3 can recover the block $W_{1,\{3\}} + W_{2,\{3\}}$; from $X_{\{1,3\}}$, user 3 can recover the block $W_{1,\{1\}} + W_{2,\{1\}}$; from $X_{\{2,3\}}$, user 3 can recover the block $W_{1,\{2\}} + W_{2,\{2\}}$. Hence, user 3 lacks the block $W_{1,\{4\}} + W_{2,\{4\}}$. If we assume that user 3 can recover the block $W_{1,\{4\}} + W_{2,\{4\}}$ from (38) and its cache, we have (39), shown at the bottom of the next page, where in (39b) we removed the cached content of user 3 in all the multicast messages, and (39c) follows since the symbols in the files are independent.

We define $(a)_b$ as a vector with length b where the elements are all a. We first let

$$W_{i,\{j\}} = (0)_{\mathsf{F}/4}, \quad \forall i \in [2], \quad j \in \{1, 2, 4\}. \tag{40}$$

With the above values of subfiles, in (39c) we have

$$X_{\{1,2\}} = W_{1,\{1\}} + W_{2,\{1\}} = X_{\{1,4\}} = W_{1,\{2\}} + W_{2,\{2\}} = X_{\{2,4\}} = (0)_{\mathsf{F}/4}. \tag{41}$$

We also have

$$W_{1,\{4\}} + W_{2,\{4\}} = (0)_{\mathsf{F}/4}. \tag{42}$$

We then let

$$W_{1,\{1\}} = (1)_{\mathsf{F}/4},\ W_{1,\{2\}} = (1)_{\mathsf{F}/4},\ W_{1,\{4\}} = (-1)_{\mathsf{F}/4}, \tag{43a}$$
$$W_{2,\{1\}} = (-1)_{\mathsf{F}/4},\ W_{2,\{2\}} = (-1)_{\mathsf{F}/4},\ W_{2,\{4\}} = (-1)_{\mathsf{F}/4}, \tag{43b}$$

With the values of subfiles in (43), we can see that (41) still holds; however, we have

$$W_{1,\{4\}} + W_{2,\{4\}} = (-1-1)_{\mathsf{F}/4}, \tag{44}$$

which is not the same as (42) if the characteristic of $\mathbb{F}_{\mathsf{q}}$ is larger than 2. As a result, we prove that the conditional entropy in (39c) is not equal to 0, which contradicts the fact that we assumed that user 3 could recover $W_{1,\{4\}} + W_{2,\{4\}}$. So we proved that if we choose the encoding coefficients as the YMA scheme, the resulting scheme is not decodable. □

B. General Description

We use the file split in (16)-(17), resulting in the demand split in (18). The placement phase is as for the MAN scheme. In the delivery phase, after the demand matrix $\mathbb{D} = \left[y_{k,n} : k \in [\mathsf{K}], n \in [\mathsf{N}]\right] \in \mathbb{F}_{\mathsf{q}}^{\mathsf{K}\times\mathsf{N}}$ is revealed, among the K users we first choose $\text{rank}_{\mathsf{q}}(\mathbb{D})$ leaders and we let $\mathcal{L}$ to be the set of the leaders, that is

$$|\mathcal{L}| = \text{rank}_{\mathsf{q}}(\mathbb{D}_{\mathcal{L}}) = \text{rank}_{\mathsf{q}}(\mathbb{D}). \tag{45}$$

Without loss of generality, that is, up to a permutation of the user indices, we can assume that $\mathcal{L} = [|\mathcal{L}|]$. This convention greatly simplifies the notation in the following of this section.

From (45), we can express the demands of non-leaders by the linear combinations of the demands of leaders. More precisely, we define that

$$B_i := y_{i,1}W_1 + \ldots + y_{i,\mathsf{N}}W_{\mathsf{N}}, \quad \forall i \in [|\mathcal{L}|], \tag{46}$$

and express the demand of each user $k \in [\mathsf{K}]$ as

$$y_{k,1}W_1 + \ldots + y_{k,\mathsf{N}}W_{\mathsf{N}} = \sum_{\ell\in\mathcal{L}} x_{k,\ell}B_\ell. \tag{47}$$

For each $k \in [\mathsf{K}]$, we define the row vector

$$\mathbf{x}_k := [x_{k,1}, \ldots, x_{k,|\mathcal{L}|}]. \tag{48}$$

Clearly, for each leader $i \in [|\mathcal{L}|]$, $\mathbf{x}_i$ is an $|\mathcal{L}|$-dimension unit vector where the ith element is 1. The transformed demand matrix $\mathbb{D}'$ is defined as follows,

$$\mathbb{D}' = [x_{1,1}, \ldots, x_{1,|\mathcal{L}|}; \ldots; x_{\mathsf{K},1}, \ldots, x_{\mathsf{K},|\mathcal{L}|}]. \tag{49}$$

In addition, for each $i \in [\mathsf{K}]$ and each $\mathcal{T} \subseteq [\mathsf{K}]$ where $|\mathcal{T}| = t$, we define the block as in (18),

$$B_{i,\mathcal{T}} := y_{i,1} W_{i,\mathcal{T}} + \ldots + y_{i,\mathsf{N}} W_{\mathsf{N},\mathcal{T}}. \tag{50}$$

If one block in (50) is demanded by a leader (i.e., $B_{i,\mathcal{T}}$ where $i \in [|\mathcal{L}|]$), we also call it a *leader block*. From (47), for each $k \in [\mathsf{K}]$ and each $\mathcal{T} \subseteq [\mathsf{K}]$ where $|\mathcal{T}| = t$, we can re-express each block $B_{k,\mathcal{T}}$ as a linear combination of leader blocks,

$$B_{k,\mathcal{T}} = y_{k,1} W_{1,\mathcal{T}} + \ldots + y_{k,\mathsf{N}} W_{\mathsf{N},\mathcal{T}} \tag{51a}$$

$$= x_{k,1} B_{1,\mathcal{T}} + \ldots + x_{k,|\mathcal{L}|} B_{|\mathcal{L}|,\mathcal{T}}. \tag{51b}$$

Encoding. For each $\mathcal{S} \subseteq [\mathsf{K}]$, we denote the set of leaders in $\mathcal{S}$ by

$$\mathcal{L}_{\mathcal{S}} := \mathcal{S} \cap \mathcal{L} = \mathcal{S} \cap [|\mathcal{L}|], \tag{52}$$

and the set of non-leaders in $\mathcal{S}$ by

$$\mathcal{N}_{\mathcal{S}} := \mathcal{S} \setminus \mathcal{L} = \mathcal{S} \setminus [|\mathcal{L}|]. \tag{53}$$

Now we focus on each set $\mathcal{S} \subseteq [\mathsf{K}]$ where $|\mathcal{S}| = t+1$, and generate the multicast message (recall that as defined in Section I-B, for any set $\mathcal{V}$, $\mathcal{V}(i)$ represents the ith smallest element in set $\mathcal{V}$)

$$X_{\mathcal{S}} = \sum_{i \in [|\mathcal{L}_{\mathcal{S}}|]} (-1)^{i-1} B_{\mathcal{L}_{\mathcal{S}}(i), \mathcal{S} \setminus \{\mathcal{L}_{\mathcal{S}}(i)\}} + \sum_{j \in [|\mathcal{N}_{\mathcal{S}}|]} (-1)^{j-1} B_{\mathcal{N}_{\mathcal{S}}(j), \mathcal{S} \setminus \{\mathcal{N}_{\mathcal{S}}(j)\}}. \tag{54}$$

The construction of $X_{\mathcal{S}}$ can be explained as follows.

- The encoding coefficient of each block in $X_{\mathcal{S}}$ is either 1 or -1.
- We divide the blocks in $X_{\mathcal{S}}$ into two groups, demanded by leaders and non-leaders, respectively. We alternate the sign (i.e., the coefficient 1 or -1) of each block demanded by leaders, and alternate the sign of each block demanded by non-leaders, respectively. We then sum the resulting summations of these two groups.

Delivery. The server broadcasts $X_{\mathcal{S}}$ for each $\mathcal{S} \subseteq [\mathsf{K}]$ where $|\mathcal{S}| = t+1$ and $\mathcal{S} \cap \mathcal{L} \neq \emptyset$.

Decoding. We consider each set $\mathcal{A} \subseteq [\mathsf{K}]$ where $|\mathcal{A}| = t+1$ and $\mathcal{A} \cap \mathcal{L} = \emptyset$.

We define that the *non-leader index* of non-leader $\mathcal{A}(i)$ is i, where $i \in [t+1]$. For each $\mathcal{S} \subseteq \mathcal{A} \cup \mathcal{L}$, recall that $\mathcal{N}_{\mathcal{S}}$ defined in (53) represents the set of non-leaders in $\mathcal{S}$. By definition, we have $\mathcal{N}_{\mathcal{S}} \subseteq \mathcal{A}$. In addition, with a slight abuse of notation we denote the non-leader indices of non-leaders in $\mathcal{A} \setminus \mathcal{S}$ by

$$\overline{\text{Ind}}_{\mathcal{S}} = \{i \in [t+1] : \mathcal{A}(i) \notin \mathcal{S}\}. \tag{55}$$

For example, if $\mathcal{A} = \{4,5,6\}$ and $\mathcal{S} = \{1,2,5\}$, we have $\overline{\text{Ind}}_{\mathcal{S}} = \{1,3\}$.

For any set $\mathcal{X}$, we define $\text{Tot}(\mathcal{X})$ as the sum of the elements in $\mathcal{X}$, i.e.,

$$\text{Tot}(\mathcal{X}) := \sum_{i \in [|\mathcal{X}|]} \mathcal{X}(i);. \tag{56}$$

For example, if $\mathcal{X} = \{1,3,4,5\}$, we have $\text{Tot}(\mathcal{X}) = 1+3+4+5 = 13$.

Recall that $\mathbb{A}_{\mathcal{S},\mathcal{V}}$ represents the sub-matrix of $\mathbb{A}$ by selecting from $\mathbb{A}$, the rows with indices in $\mathcal{S}$ and the columns with indices in $\mathcal{V}$. It will be proved in Appendix B (which is the most technical part of this article) that

$$X_{\mathcal{A}} = \sum_{\mathcal{S} \subseteq \mathcal{A} \cup \mathcal{L} : |\mathcal{S}| = t+1, \mathcal{S} \neq \mathcal{A}} \beta_{\mathcal{A},\mathcal{S}} X_{\mathcal{S}}, \tag{57a}$$

$$\beta_{\mathcal{A},\mathcal{S}} = (-1)^{1+\text{Tot}(\overline{\text{Ind}}_{\mathcal{S}})} \det(\mathbb{D}'_{\mathcal{A}\setminus\mathcal{S},\mathcal{L}_{\mathcal{S}}}). \tag{57b}$$

In other words, each user $k \in [\mathsf{K}]$ can recover all messages $X_{\mathcal{S}}$ where $\mathcal{S} \subseteq [\mathsf{K}]$ and $|\mathcal{S}| = t+1$. For each desired block $B_{k,\mathcal{T}}$, where $\mathcal{T} \subseteq ([\mathsf{K}] \setminus \{k\})$ and $|\mathcal{T}| = t$, user k can recover it in $X_{\mathcal{T} \cup \{k\}}$, because it knows all the other blocks in $X_{\mathcal{T} \cup \{k\}}$. Hence, user k can recover $x_{k,1} B_1 + \ldots + x_{k,|\mathcal{L}|} B_{|\mathcal{L}|}$, which is identical to its demand.

Performance. In total, we transmit $\binom{\mathsf{K}}{t+1} - \binom{\mathsf{K}-\text{rank}_{\mathsf{q}}(\mathbb{D})}{t+1}$ multicast messages, each of which contains $\frac{\mathsf{F}}{\binom{\mathsf{K}}{t}}$ symbols. Hence, the transmitted load is

$$\frac{\binom{\mathsf{K}}{t+1} - \binom{\mathsf{K}-\text{rank}_{\mathsf{q}}(\mathbb{D})}{t+1}}{\binom{\mathsf{K}}{t}}. \tag{58}$$

For the worst-case demands where $\text{rank}_{\mathsf{q}}(\mathbb{D})$ is full-rank, we have $\text{rank}_{\mathsf{q}}(\mathbb{D}) = \min\{\mathsf{K}, \mathsf{N}\}$, and we achieve the worst-case load in (22).

C. Simplified Proof of Decodability for $\mathsf{q} = 2$

When $\mathsf{q} = 2$, inspired from the YMA scheme's decoding step reviewed in Section II-B, we can give an alternate decodability proof for the proposed cache-aided function retrieval scheme, which only needs sum/XOR operations and is thus much simpler than the one for the general q in Appendix B. We also start with an example.

Example 1: Consider the $(\mathsf{K}, \mathsf{N}, \mathsf{q}) = (6,3,2)$ shared-link cache-aided scalar linear function retrieval problem, where

$$0 = H\big(W_{1,\{4\}} + W_{2,\{4\}} \big| X_{\{1,2\}}, X_{\{1,3\}}, X_{\{1,4\}} X_{\{2,3\}}, X_{\{2,4\}}, \{W_{i,\{3\}} : i \in [2]\}\big) \tag{39a}$$

$$= H\big(W_{1,\{4\}} + W_{2,\{4\}} \big| X_{\{1,2\}}, W_{1,\{1\}} + W_{2,\{1\}}, X_{\{1,4\}}, W_{1,\{2\}} + W_{2,\{2\}}, X_{\{2,4\}}, \{W_{i,\{3\}} : i \in [2]\}\big) \tag{39b}$$

$$= H\big(W_{1,\{4\}} + W_{2,\{4\}} \big| X_{\{1,2\}}, W_{1,\{1\}} + W_{2,\{1\}}, X_{\{1,4\}}, W_{1,\{2\}} + W_{2,\{2\}}, X_{\{2,4\}}\big). \tag{39c}$$

$t = \mathsf{KM}/\mathsf{N} = 2$, that is, $\mathsf{M} = 1$. In the cache placement, each file is partitioned into $\binom{\mathsf{K}}{t} = 15$ equal-length subfiles. We use the file split in (16)-(17), resulting in the demand split in (18). In the delivery phase, we assume that the demand matrix is

$$\mathbb{D} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}. \tag{59}$$

On the observation that $\text{rank}_2(\mathbb{D}) = 3$, we choose 3 users as leaders, where the demand matrix of these 3 leaders is also full-rank. Here, we choose $\mathcal{L} = [3]$.

Encoding. For each set $\mathcal{S} \subseteq [\mathsf{K}]$ where $|\mathcal{S}| = t+1 = 3$, we generate a multicast message with encoding coefficients equal to 1 in (19). Hence, we have

$$X_{\{1,2,3\}} = W_{1,\{2,3\}} \oplus W_{2,\{1,3\}} \oplus W_{3,\{1,2\}}; \tag{60a}$$
$$X_{\{1,2,4\}} = W_{1,\{2,4\}} \oplus W_{2,\{1,4\}} \oplus (W_{1,\{1,2\}} \oplus W_{2,\{1,2\}}); \tag{60b}$$
$$X_{\{1,2,5\}} = W_{1,\{2,5\}} \oplus W_{2,\{1,5\}} \oplus (W_{1,\{1,2\}} \oplus W_{3,\{1,2\}}); \tag{60c}$$
$$X_{\{1,2,6\}} = W_{1,\{2,6\}} \oplus W_{2,\{1,6\}} \oplus (W_{1,\{1,2\}} \oplus W_{2,\{1,2\}} \oplus W_{3,\{1,2\}}); \tag{60d}$$
$$X_{\{1,3,4\}} = W_{1,\{3,4\}} \oplus W_{3,\{1,4\}} \oplus (W_{1,\{1,3\}} \oplus W_{2,\{1,3\}}); \tag{60e}$$
$$X_{\{1,3,5\}} = W_{1,\{3,5\}} \oplus W_{3,\{1,5\}} \oplus (W_{1,\{1,3\}} \oplus W_{3,\{1,3\}}); \tag{60f}$$
$$X_{\{1,3,6\}} = W_{1,\{3,6\}} \oplus W_{3,\{1,6\}} \oplus (W_{1,\{1,3\}} \oplus W_{2,\{1,3\}} \oplus W_{3,\{1,3\}}); \tag{60g}$$
$$X_{\{1,4,5\}} = W_{1,\{4,5\}} \oplus (W_{1,\{1,5\}} \oplus W_{2,\{1,5\}}) \oplus (W_{1,\{1,4\}} \oplus W_{3,\{1,4\}}); \tag{60h}$$
$$X_{\{1,4,6\}} = W_{1,\{4,6\}} \oplus (W_{1,\{1,6\}} \oplus W_{2,\{1,6\}}) \oplus (W_{1,\{1,4\}} \oplus W_{2,\{1,4\}} \oplus W_{3,\{1,4\}}); \tag{60i}$$
$$X_{\{1,5,6\}} = W_{1,\{5,6\}} \oplus (W_{1,\{1,6\}} \oplus W_{3,\{1,6\}}) \oplus (W_{1,\{1,5\}} \oplus W_{2,\{1,5\}} \oplus W_{3,\{1,5\}}); \tag{60j}$$
$$X_{\{2,3,4\}} = W_{2,\{3,4\}} \oplus W_{3,\{2,4\}} \oplus (W_{1,\{2,3\}} \oplus W_{2,\{2,3\}}); \tag{60k}$$
$$X_{\{2,3,5\}} = W_{2,\{3,5\}} \oplus W_{3,\{2,5\}} \oplus (W_{1,\{2,3\}} \oplus W_{3,\{2,3\}}); \tag{60l}$$
$$X_{\{2,3,6\}} = W_{2,\{3,6\}} \oplus W_{3,\{2,6\}} \oplus (W_{1,\{2,3\}} \oplus W_{2,\{2,3\}} \oplus W_{3,\{2,3\}}); \tag{60m}$$
$$X_{\{2,4,5\}} = W_{2,\{4,5\}} \oplus (W_{1,\{2,5\}} \oplus W_{2,\{2,5\}}) \oplus (W_{1,\{2,4\}} \oplus W_{3,\{2,4\}}); \tag{60n}$$
$$X_{\{2,4,6\}} = W_{2,\{4,6\}} \oplus (W_{1,\{2,6\}} \oplus W_{2,\{2,6\}}) \oplus (W_{1,\{2,4\}} \oplus W_{2,\{2,4\}} \oplus W_{3,\{2,4\}}); \tag{60o}$$
$$X_{\{2,5,6\}} = W_{2,\{5,6\}} \oplus (W_{1,\{2,6\}} \oplus W_{3,\{2,6\}}) \oplus (W_{1,\{2,5\}} \oplus W_{2,\{2,5\}} \oplus W_{3,\{2,5\}}); \tag{60p}$$
$$X_{\{3,4,5\}} = W_{3,\{4,5\}} \oplus (W_{1,\{3,5\}} \oplus W_{2,\{3,5\}}) \oplus (W_{1,\{3,4\}} \oplus W_{3,\{3,4\}}); \tag{60q}$$
$$X_{\{3,4,6\}} = W_{3,\{4,6\}} \oplus (W_{1,\{3,6\}} \oplus W_{2,\{3,6\}}) \oplus (W_{1,\{3,4\}} \oplus W_{2,\{3,4\}} \oplus W_{3,\{3,4\}}); \tag{60r}$$
$$X_{\{3,5,6\}} = W_{3,\{5,6\}} \oplus (W_{1,\{3,6\}} \oplus W_{3,\{3,6\}}) \oplus (W_{1,\{3,5\}} \oplus W_{2,\{3,5\}} \oplus W_{3,\{3,5\}}); \tag{60s}$$
$$X_{\{4,5,6\}} = (W_{1,\{5,6\}} \oplus W_{2,\{5,6\}}) \oplus (W_{1,\{4,6\}} \oplus W_{3,\{4,6\}}) \oplus (W_{1,\{4,5\}} \oplus W_{2,\{4,5\}} \oplus W_{3,\{4,5\}}). \tag{60t}$$

Delivery. The server broadcasts $X_{\mathcal{S}}$ for each $\mathcal{S} \subseteq [\mathsf{K}]$ where $|\mathcal{S}| = t+1 = 3$ and $\mathcal{S} \cap \mathcal{L} \neq \emptyset$. In other words, the server broadcasts all the multicast messages in (60) except for $X_{\{4,5,6\}}$.

Decoding. We show that the untransmitted multicast message $X_{\{4,5,6\}}$ can be reconstructed by the transmitted multicast messages. For each set of users $\mathcal{B} \subseteq [\mathsf{K}]$, we define $\mathscr{V}_{\mathcal{B}}$ as the family of subsets $\mathcal{V} \subseteq \mathcal{B}$, where $|\mathcal{V}| = |\mathcal{L}|$ and $\text{rank}_2(\mathbb{D}_{\mathcal{V}}) = |\mathcal{L}|$. It can be seen that $\mathscr{V}_{\mathcal{B}}$ is the generalization of $\mathscr{F}_{\mathcal{B}}$ defined in the YMA scheme described in Section II-B. When $\mathcal{B} = \mathcal{L} \cup \{4,5,6\} = [6]$, we have

$$\mathscr{V}_{[6]} = \Big\{\{1,2,3\}, \{1,2,5\}, \{1,2,6\}, \{1,3,4\}, \{1,3,6\}, \{1,4,5\}, \{1,4,6\}, \{1,5,6\}, \{2,3,4\}, \{2,3,5\}, \{2,3,6\}, \{2,4,5\}, \{2,4,6\}, \{3,4,5\}, \{3,5,6\}, \{4,5,6\}\Big\}. \tag{61}$$

From the above definition, we focus on the following sum of multicast messages

$$\bigoplus_{\mathcal{V} \in \mathscr{V}_{[6]}} X_{[6]\setminus\mathcal{V}} = 0, \tag{62}$$

where (62) is because on the LHS of (62), among all subfiles $W_{i,\mathcal{T}}$ where $i \in [3]$, $\mathcal{T} \subseteq [6]$, and $|\mathcal{T}| = 2$, the coefficient of each of $W_{2,\{2,4\}}$, $W_{2,\{2,6\}}$, $W_{2,\{4,6\}}$, $W_{3,\{3,5\}}$, $W_{3,\{3,6\}}$, $W_{3,\{5,6\}}$ is 0, $W_{1,\{2,3\}}$ appears 4 times and other subfiles appear 2 times. Hence, the sum is equal to 0 on $\mathbb{F}_2$. Note that in the YMA scheme, each subfile appears 2 times in the sum $\bigoplus_{\mathcal{F} \in \mathscr{F}_{\mathcal{B}}} X_{\mathcal{B}\setminus\mathcal{F}}$.

We can write (62) as

$$X_{\{4,5,6\}} = \bigoplus_{\mathcal{V} \in \mathscr{V}_{[6]}: \mathcal{V} \neq \mathcal{L}} X_{[6]\setminus\mathcal{V}}. \tag{63}$$

In other words, the untransmitted multicast message $X_{\{4,5,6\}}$ can be reconstructed by the transmitted multicast messages. Thus each user can recover all the multicast messages in (60), and then recover its desired function.

Performance. In total we transmit $\binom{\mathsf{K}}{t+1} - \binom{\mathsf{K}-\text{rank}_2(\mathbb{D})}{t+1} = \binom{6}{3} - \binom{3}{3} = 19$ multicast messages, each of which contains $\frac{\mathsf{F}}{20}$ bits. Hence, the transmitted load is $\frac{19}{20}$, which coincides with the optimal worst-case load in Theorem 2. □

Remark 5: In Example 1, if we change the finite field from $\mathbb{F}_2$ to $\mathbb{F}_4$, and consider the demands of users $\mathsf{y}_1 = (1,0,0)$, $\mathsf{y}_2 = (0,1,0)$, $\mathsf{y}_3 = (0,0,1)$, $\mathsf{y}_4 = (1,2,0)$, $\mathsf{y}_5 = (1,0,1)$, and $\mathsf{y}_6 = (1,2,1)$. The encoding coefficients are still equal to 1, because we also have $+1 = -1 = 1$ on $\mathbb{F}_4$. However, the YMA decoding procedure cannot work. More precisely, in this case $\mathscr{V}_{[6]}$ is the same as (61). If we take the sum in (62)

following the YMA decoding procedure, some subfiles such as $W_{2,\{1,3\}}$ appear twice in (62), where $X_{1,2,3} = W_{1,\{2,3\}} + W_{2,\{1,3\}} + W_{3,\{1,2\}}$ and $X_{1,3,6} = W_{1,\{3,6\}} + W_{3,\{1,6\}} + (W_{1,\{1,3\}} + 2W_{2,\{1,3\}} + W_{3,\{1,3\}})$; thus the coefficient of $W_{2,\{1,3\}}$ in (62) is 3, not 0 on $\mathbb{F}_4$. In this case, multiplication operations are needed during the decoding step. □

For the binary field $\mathbb{F}_2$, we are now ready to generalize the decodability proof in the above example. Recall that in the delivery phase, among the K users we first choose $\text{rank}_2(\mathbb{D})$ leaders, where

$$|\mathcal{L}| = \text{rank}_2(\mathbb{D}_{\mathcal{L}}) = \text{rank}_2(\mathbb{D}). \tag{64}$$

For each set $\mathcal{S} \subseteq [\mathsf{K}]$ where $|\mathcal{S}| = t+1$, we generate the multicast messages

$$X_{\mathcal{S}} = \underset{k\in\mathcal{S}}{\oplus} B_{k,\mathcal{S}\setminus\{k\}},\ \forall \mathcal{S} \subseteq [\mathsf{K}] : |\mathcal{S}| = t+1, \tag{65}$$

where the encoding coefficient of each $B_{k,\mathcal{S}\setminus\{k\}}$ is 1. The server then sends the multicast messages $X_{\mathcal{S}}$ where $|\mathcal{S}| = t+1$ and $\mathcal{S} \cap \mathcal{L} \neq \emptyset$.

For each set of users $\mathcal{B} \subseteq [\mathsf{K}]$, define $\mathscr{V}_{\mathcal{B}}$ as the family of subsets $\mathcal{V} \subseteq \mathcal{B}$, where $|\mathcal{V}| = |\mathcal{L}|$ and $\text{rank}_2(\mathbb{D}_{\mathcal{V}}) = |\mathcal{L}|$. We now consider each set $\mathcal{A} \subseteq [\mathsf{K}]$ where $|\mathcal{A}| = t+1$ and $\mathcal{A} \cap \mathcal{L} = \emptyset$, and focus on the binary sum with $\mathcal{B} = \mathcal{L} \cup \mathcal{A}$,

$$\underset{\mathcal{V}\in\mathscr{V}_{\mathcal{B}}}{\oplus} X_{\mathcal{B}\setminus\mathcal{V}}. \tag{66}$$

A subfile $W_{i,\mathcal{T}}$ where $\mathcal{T} \subseteq \mathcal{B}$ appears in the sum (66) if and only if there exists some user $k \in \mathcal{B} \setminus \mathcal{T}$ such that $\text{rank}_2(\mathbb{D}_{\mathcal{B}\setminus(\mathcal{T}\cup\{k\})}) = |\mathcal{L}|$ (i.e., $\mathbb{D}_{\mathcal{B}\setminus(\mathcal{T}\cup\{k\})}$ is full-rank and thus $X_{\mathcal{T}\cup\{k\}}$ is in (66)) and $y_{k,i} \neq 0$ (i.e., $W_{i,\mathcal{T}}$ appears in $X_{\mathcal{T}\cup\{k\}}$). We then provide the following Lemma, proved in Appendix A.

Lemma 1: If $W_{i,\mathcal{T}}$ appears in the sum (66), the number of multicast messages in the sum which contains $W_{i,\mathcal{T}}$ is even.[5] □

From Lemma 1, it can be seen that each subfile in the sum (66) appears an even number of times, and thus the coefficient of this subfile in the sum is 0, which leads to

$$X_{\mathcal{A}} = \underset{\mathcal{V}\in\mathscr{V}_{\mathcal{B}}:\mathcal{V}\neq\mathcal{L}}{\oplus} X_{\mathcal{B}\setminus\mathcal{V}}. \tag{67}$$

In other words, $X_{\mathcal{A}}$ can be reconstructed by the transmitted multicast messages.

As a result, each user can recover all multicast messages and thus it can decode its desired function.

V. Conclusion

In this article, we introduced the novel problem of cache-aided scalar linear function retrieval, which is a generalization of the classic coded caching problem and allows users to request scalar linear functions of files. We proposed a novel scheme that is provably correct for demand functions over arbitrary finite field. Quite surprisingly, the proposed scheme has the same worst-case load performance as the optimal scheme under uncoded cache placement for single file retrieval, as such it is optimal under the constraint of uncoded cache placement and to within a factor of two otherwise. Ongoing work includes the extension of the proposed caching scheme to the case where the demanded functions are vector linear or non-linear, as well as finding novel caching schemes for the cache-aided function retrieval problem with coded cache placement.

[5] Note that in the YMA scheme for the original MAN caching problem, each subfile in (11) is contained in two multicast messages in (11). Hence, Lemma 1 is also a generalization of [4, Lemma 1] for the YMA scheme.

Appendix A
Proof of Lemma 1

Recall that a subfile $W_{i,\mathcal{T}}$ where $\mathcal{T} \subseteq \mathcal{B}$ appears in the sum (66) if and only if there exists some user $k \in \mathcal{B} \setminus \mathcal{T}$ satisfying the following two constraints,

1) Constraint 1: $\text{rank}_2(\mathbb{D}_{\mathcal{B}\setminus(\mathcal{T}\cup\{k\})}) = |\mathcal{L}|$ such that $X_{\mathcal{T}\cup\{k\}}$ is in (66);
2) Constraint 2: $y_{k,i} \neq 0$ such that $W_{i,\mathcal{T}}$ appears in $X_{\mathcal{T}\cup\{k\}}$.

Thus, for each $k \in \mathcal{B} \setminus \mathcal{T}$ satisfying the above constraints, in (66) there exists one multicast message which contains $W_{i,\mathcal{T}}$. So to prove Lemma 1, it is equivalent to prove that the number of users $k \in \mathcal{B} \setminus \mathcal{T}$ satisfying the above constraints is even.

We assume that user k_1 satisfies the above constraints. Hence, $\mathbb{D}_{\mathcal{B}\setminus(\mathcal{T}\cup\{k_1\})}$ is full-rank, and $y_{k_1,i} \neq 0$. We let $\mathcal{Y} = \{\mathcal{Y}(1), \ldots, \mathcal{Y}(|\mathcal{L}|)\} = \mathcal{B} \setminus (\mathcal{T} \cup \{k_1\})$.

In the following, we operate a linear space transformation. More precisely, we let

$$G_j = \mathbf{y}_{\mathcal{Y}(j)}[W_1; \ldots; W_{\mathsf{N}}], \quad \forall j \in [|\mathcal{L}|]. \tag{68}$$

From (68), we can re-write the demand of each user $\mathcal{Y}(j)$ as

$$G_j = \mathbf{y}'_j[G_1; \ldots; G_{|\mathcal{L}|}],$$

where $\mathbf{y}'_j$ is the $|\mathcal{L}|$-dimension unit vector whose j^{th} element is 1. The transformed demand matrix of the users in $\mathcal{Y}$ is

$$\mathbb{T} = [\mathbf{y}'_1; \ldots; \mathbf{y}'_{|\mathcal{L}|}],$$

which is an identity matrix.

In addition, we can also re-write the demand of user k_1 as

$$\mathbf{y}'[G_1; \ldots; G_{|\mathcal{L}|}],$$

where $\mathbf{y}'$ is an $|\mathcal{L}|$-dimension vector on $\mathbb{F}_2$. Note that if the pth element in $\mathbf{y}'$ is 1 and G_p contains W_i, W_i appears one time in $\mathbf{y}'[G_1; \ldots; G_{|\mathcal{L}|}]$. As $y_{k_1,i} \neq 0$, it can be seen that $\mathbf{y}'[G_1; \ldots; G_{|\mathcal{L}|}]$ contains W_i. Hence, the number of $p \in [|\mathcal{L}|]$ where the pth element in $\mathbf{y}'$ is 1 and G_p contains W_i, is odd. For each of such p, if we replace the pth row of $\mathbb{T}$ by $\mathbf{y}'$, the resulting matrix is still full-rank, because the pth element in $\mathbf{y}'$ is 1. As the resulting matrix is full-rank, it can be seen that $\mathbb{D}_{\mathcal{B}\setminus(\mathcal{T}\cup\{\mathcal{Y}(p)\})}$ is also full-rank. In addition, as G_p contains W_i, we can see that $y_{\mathcal{Y}(p),i} \neq 0$. Hence, user $\mathcal{Y}(p)$ also satisfies the two constraints. Moreover, for any $s \in [|\mathcal{L}|]$, if the sth element in $\mathbf{y}'$ is not 1, user $\mathcal{Y}(s)$ does not satisfy Constraint 1; if G_s does not contain W_s, user $\mathcal{Y}(s)$ does not satisfy Constraint 2.

As a result, besides user k_1, the number of users in $\mathcal{B} \setminus \mathcal{T}$ satisfying the two constraints is odd. In conclusion, by taking user k_1 into consideration, the number of users in $\mathcal{B} \setminus \mathcal{T}$

satisfying the two constraints is even. Thus Lemma 1 is proved.

APPENDIX B
PROOF OF (57)

Recall that $\mathcal{L} = [|\mathcal{L}|]$. We now focus on one set of non-leaders $\mathcal{A} \subseteq [\mathsf{K}]$ where $|\mathcal{A}| = t+1$ and $\mathcal{A} \cap \mathcal{L} = \emptyset$.

For any positive integer n, $\text{Perm}(n)$ represents the set of all permutations of $[n]$. For any set $\mathcal{X}$ and any number y, we define $\text{Card}(\mathcal{X}, y)$ as the number of elements in $\mathcal{X}$ which is smaller than y, i.e.,

$$\text{Card}(\mathcal{X}, y) := |\{i \in \mathcal{X} : i < y\}|. \tag{69}$$

For example, if $\mathcal{X} = \{1, 3, 4, 5\}$ and $y = 4$, we have $\text{Card}(\mathcal{X}, y) = |\{1, 3\}| = 2$.

Our objective is to prove (57) which can be equivalently written as

$$X_{\mathcal{A}} = \sum_{\mathcal{S} \subseteq \mathcal{A} \cup \mathcal{L} : |\mathcal{S}| = t+1, \mathcal{S} \neq \mathcal{A}} \beta_{\mathcal{A},\mathcal{S}} X_{\mathcal{S}}, \tag{70}$$

where $\beta_{\mathcal{A},\mathcal{S}}$ is given in (71), shown at the bottom of the next page. Note that (71) is obtained from expanding the determinant in (57b).

Remark 6 (Connection to the Example in Section IV-A – Part I): Let us go back to the illustrated example in Section IV-A, where we have $\mathcal{L} = [2]$. When $\mathcal{A} = \{3, 4\}$ and $\mathcal{S} = \{1, 2\}$, from the definition in (52) we have $\mathcal{L}_{\mathcal{S}} = [2]$ and from the definition in (55) we have $\overline{\text{Ind}}_{\mathcal{S}} = [2]$. In addition, $\text{Perm}(|\mathcal{L}_{\mathcal{S}}|) = \text{Perm}(2) = \{(1, 2), (2, 1)\}$. Hence, when $\mathbf{u} = (u_1, u_2) = (1, 2)$, in (71) we have the term

$$\begin{aligned}
&(-1)^{1+\text{Tot}([2])+\text{Card}([2]\setminus\{1\},1)+\text{Card}([2]\setminus\{1,2\},2)} x_{3,1} x_{4,2} \\
&= (-1)^{1+\text{Tot}([2])+\text{Card}([2]\setminus\{1\},1)+\text{Card}([2]\setminus\{1,2\},2)} y_{3,1} y_{4,2} \\
&= y_{3,1} y_{4,2},
\end{aligned} \tag{72}$$

where (72) is because in the example we have $W_i = B_i$ for each $i \in [\mathsf{N}]$, and thus $\mathbf{x}_k = \mathbf{y}_k$ for each $k \in [\mathsf{K}]$. Similarly, when $\mathbf{u} = (2, 1)$, in (71) we have the term

$$\begin{aligned}
&(-1)^{1+\text{Tot}([2])+\text{Card}([2]\setminus\{2\},2)+\text{Card}([2]\setminus\{1,2\},1)} x_{4,1} x_{3,2} \\
&= (-1)^{1+\text{Tot}([2])+\text{Card}([2]\setminus\{2\},2)+\text{Card}([2]\setminus\{1,2\},1)} y_{4,1} y_{3,2} \\
&= -y_{4,1} y_{3,2}.
\end{aligned} \tag{73}$$

Hence, in (71) we have $\beta_{\{3,4\},\{1,2\}} = y_{3,1} y_{4,2} - y_{4,1} y_{3,2}$, which coincides with (37a) in the illustrated example. □

By the definition of $X_{\mathcal{S}}$ in (54) and the fact that each block is a linear combination of leader blocks as explained in (51b), it is obvious to check that in (70), there only exist the leader blocks $B_{i,\mathcal{T}}$ where $i \in [|\mathcal{L}|]$, $\mathcal{T} \subseteq (\mathcal{A} \cup \mathcal{L})$, and $|\mathcal{T}| = t$. Now we divide such leader blocks into hierarchies, where we say a leader block $B_{i,\mathcal{T}}$ appearing in (70) is in Hierarchy $h \in [0 : t]$, if $|\mathcal{T} \cap \mathcal{L}| = h$. In addition, on the LHS of (70), only leader blocks in Hierarchy 0 exist.

We consider the following three cases:

1) Case 1: $B_{i,\mathcal{T}}$ is in Hierarchy 0. In Appendix B-A, we will prove that the coefficient of $B_{i,\mathcal{T}}$ on the RHS of (70) is equal to the coefficient of $B_{i,\mathcal{T}}$ on the LHS of (70).
2) Case 2: $B_{i,\mathcal{T}}$ is in Hierarchy $h > 0$ and $i \in \mathcal{T}$. In Appendix B-B, we will prove that the coefficient of $B_{i,\mathcal{T}}$ on the RHS of (70) is 0.
3) Case 3: $B_{i,\mathcal{T}}$ is in Hierarchy $h > 0$ and $i \notin \mathcal{T}$. In Appendix B-C, we will prove that the coefficient of $B_{i,\mathcal{T}}$ on the RHS of (70) is 0.

Hence, after proving the above three cases, (70) can be directly derived.

Remark 7 (Connection to the Example in Section IV-A – part II): In the illustrated example, recall that $B_i = W_i$ for each $i \in [|\mathcal{L}|]$. Hence, it can be seen that $B_{i,\mathcal{T}} = W_{i,\mathcal{T}}$ for each $i \in [|\mathcal{L}|]$, $\mathcal{T} \subseteq [\mathsf{K}]$, and $|\mathcal{T}| = t$. For each leader block $W_{i,\mathcal{T}}$, it is in one of the following three cases,

1) Case 1: $W_{i,\mathcal{T}}$ is in Hierarchy 0. In this case, we have the leader blocks $W_{i,\{3\}}$, $W_{i,\{4\}}$ for $i \in [2]$.
2) Case 2: $W_{i,\mathcal{T}}$ is in Hierarchy 1 and $i \in \mathcal{T}$. In this case, we have the leader blocks $W_{1,\{1\}}$ and $W_{2,\{2\}}$.
3) Case 3: $W_{i,\mathcal{T}}$ is in Hierarchy 1 and $i \notin \mathcal{T}$. In this case, we have the leader blocks $W_{1,\{2\}}$ and $W_{2,\{1\}}$.

In this example, (70) becomes (29b), i.e.,

$$X_{\{3,4\}} = \sum_{\mathcal{S} \subseteq [4] : |\mathcal{S}| = 2, \mathcal{S} \cap [2] \neq \emptyset} \beta_{\{3,4\},\mathcal{S}} X_{\mathcal{S}}, \tag{74}$$

which is our objective. □

A. Case 1

If $B_{i,\mathcal{T}}$ is in Hierarchy 0, we have $\mathcal{T} \subseteq \mathcal{A}$. As $|\mathcal{A}| - |\mathcal{T}| = 1$, we assume that $\{\mathcal{A}(k)\} = \mathcal{A} \setminus \mathcal{T}$. On the LHS of (70), $B_{i,\mathcal{T}}$ appears in $X_{\mathcal{A}}$, where from (54) we have

$$\begin{aligned}
X_{\mathcal{A}} = \sum_{j \in [t+1]} (-1)^{j-1} (&x_{\mathcal{A}(j),1} B_{1,\mathcal{S}\setminus\{\mathcal{A}(j)\}} + \\
&\ldots + x_{\mathcal{A}(j),|\mathcal{L}|} B_{|\mathcal{L}|,\mathcal{S}\setminus\{\mathcal{A}(j)\}}).
\end{aligned} \tag{75}$$

Hence, the coefficient of $B_{i,\mathcal{T}}$ in $X_{\mathcal{A}}$ is $(-1)^{k-1} x_{\mathcal{A}(k),i}$.

Let us then focus on the RHS of (70). $B_{i,\mathcal{T}}$ appears in $X_{\mathcal{T}\cup\{i\}}$. As user i is the only leader in $\mathcal{T} \cup \{i\}$ (i.e., $\mathcal{L}_{\mathcal{T}\cup\{i\}} = \{i\}$), the coefficient of $B_{i,\mathcal{T}}$ in $X_{\mathcal{T}\cup\{i\}}$ is $(-1)^{1-1} = 1$. In addition, by computing $\overline{\text{Ind}}_{\mathcal{T}\cup\{i\}} = \{k\}$, we have

$$\beta_{\mathcal{A},\mathcal{T}\cup\{i\}} = (-1)^{1+k+0} x_{\mathcal{A}(k),i} = (-1)^{k+1} x_{\mathcal{A}(k),i} \tag{76a}$$

$$= (-1)^{k-1} x_{\mathcal{A}(k),i}. \tag{76b}$$

Hence, the coefficient of $B_{i,\mathcal{T}}$ on the RHS of (70) (i.e., in $\beta_{\mathcal{A},\mathcal{T}\cup\{i\}} X_{\mathcal{T}\cup\{i\}}$) is

$$(-1)^{k-1} x_{\mathcal{A}(k),i} \times 1 = (-1)^{k-1} x_{\mathcal{A}(k),i}, \tag{77}$$

which is the same as the coefficient of $B_{i,\mathcal{T}}$ on the LHS of (70).

Remark 8 (Connection to the Example in Section IV-A – Part III): In the illustrated example, recall that $\mathcal{L} = [2]$ and $\mathcal{A} = \{3, 4\}$. Let us focus on $W_{1,\{3\}}$ which is in Case 1. On the LHS of (74), we have

$$X_{\{3,4\}} = (y_{3,1} W_{1,\{4\}} + y_{3,2} W_{2,\{4\}}) - (y_{4,1} W_{1,\{3\}} + y_{4,2} W_{2,\{3\}});$$

thus the coefficient of $W_{1,\{3\}}$ in $X_{\{3,4\}}$ is $-y_{4,1}$. On the RHS of (74), $W_{1,\{3\}}$ appears in $X_{\{1,3\}}$, where

$$X_{\{1,3\}} = W_{1,\{3\}} + (y_{3,1} W_{1,\{1\}} + y_{3,2} W_{2,\{1\}});$$

thus the coefficient of $W_{1,\{3\}}$ in $X_{\{1,3\}}$ is 1. From (71) the decoding coefficient of $X_{\{1,3\}}$ is (recall that $\mathcal{L}_{\{1,3\}} = \{1\}$ and $\overline{\text{Ind}}_{\{1,3\}} = \{2\}$)

$$\beta_{\{3,4\},\{1,3\}} = \sum_{\mathbf{u}\in\text{Perm}(1)} (-1)^{1+2+\text{Card}([1]\setminus\{u_1\},u_1)} y_{4,1} = -y_{4,1}. \tag{78}$$

Hence, the coefficient of $W_{1,\{3\}}$ on the RHS of (74) is also $-y_{4,1}$, which is the same as the one on the LHS of (74). □

B. Case 2

Now we focus on one leader block $B_{i,\mathcal{T}}$ in Hierarchy $h > 0$ where $i \in \mathcal{T}$. By definition, we have $|\mathcal{T}\cap\mathcal{L}| = h$. On the RHS of (70), as $i \in \mathcal{T}$, $B_{i,\mathcal{T}}$ only appears in $X_{\mathcal{T}\cup\{\mathcal{A}(k)\}}$, where $k \in \overline{\text{Ind}}_{\mathcal{T}}$. We define that

$$\text{the } \left(\overline{\text{Ind}}_{\mathcal{T}}^{-1}(k)\right)^{\text{th}} \text{ smallest element in } \overline{\text{Ind}}_{\mathcal{T}} \text{ is } k. \tag{79}$$

We focus on one $k \in \overline{\text{Ind}}_{\mathcal{T}}$. $\mathcal{A}(k)$ is the kth element in $\mathcal{A}$, and in $\mathcal{A}\setminus\mathcal{T}$ there are $\overline{\text{Ind}}_{\mathcal{T}}^{-1}(k) - 1$ elements smaller than $\mathcal{A}(k)$. Hence, in $\mathcal{N}_{\mathcal{T}\cup\{\mathcal{A}(k)\}}$ there are $k - 1 - \left(\overline{\text{Ind}}_{\mathcal{T}}^{-1}(k) - 1\right) = k - \overline{\text{Ind}}_{\mathcal{T}}^{-1}(k)$ elements smaller than $\mathcal{A}(k)$. So from (54), it can be seen that the coefficient of $B_{i,\mathcal{T}}$ in $X_{\mathcal{T}\cup\{\mathcal{A}(k)\}}$ is

$$(-1)^{k-\overline{\text{Ind}}_{\mathcal{T}}^{-1}(k)} x_{\mathcal{A}(k),i}. \tag{80}$$

In addition, we have (81), shown at the bottom of the next page. From (80) and (81b), the coefficient of $B_{i,\mathcal{T}}$ in $\beta_{\mathcal{A},\mathcal{T}\cup\{\mathcal{A}(k)\}} X_{\mathcal{T}\cup\{\mathcal{A}(k)\}}$ is $(-1)^{k-\overline{\text{Ind}}_{\mathcal{T}}^{-1}(k)} x_{\mathcal{A}(k),i}\beta_{\mathcal{A},\mathcal{T}\cup\{\mathcal{A}(k)\}}$.

In the following, we will prove

$$\sum_{k\in\overline{\text{Ind}}_{\mathcal{T}}} (-1)^{k-\overline{\text{Ind}}_{\mathcal{T}}^{-1}(k)} x_{\mathcal{A}(k),i}\beta_{\mathcal{A},\mathcal{T}\cup\{\mathcal{A}(k)\}} = 0, \tag{82}$$

such that the coefficient of $B_{i,\mathcal{T}}$ on the RHS of (70) is 0.

Let us focus on one $k \in \overline{\text{Ind}}_{\mathcal{T}}$ and one permutation $\mathbf{u} = (u_1, \ldots, u_{|\mathcal{L}_{\mathcal{T}}|}) \in \text{Perm}(|\mathcal{L}_{\mathcal{T}}|)$. The term in (82) caused by k and $\mathbf{u}$ is (83), shown at the bottom of the next page. Note that in

$$x_{\mathcal{A}(k),i} \prod_{i_2\in[|\mathcal{L}_{\mathcal{T}}|]} x_{\mathcal{A}\left(\overline{\text{Ind}}_{\mathcal{T}\cup\{\mathcal{A}(k)\}}(u_{i_2})\right),\mathcal{L}_{\mathcal{T}}(i_2)}, \tag{84}$$

which is the product of $|\mathcal{L}_{\mathcal{T}}| + 1$ terms, there is one term whose second subscript is i' for each $i' \in \mathcal{L}_{\mathcal{T}}\setminus\{i\}$, and there are two terms whose second subscript is i. We define that

$$\text{the } \left(\mathcal{L}_{\mathcal{T}}^{-1}(i)\right)^{\text{th}} \text{ smallest element in } \mathcal{L}_{\mathcal{T}} \text{ is } i. \tag{85}$$

Hence, the two terms in (84) whose second subscript is i are $x_{\mathcal{A}(k),i}$ and $x_{\mathcal{A}(k'),i}$, where $k' := \overline{\text{Ind}}_{\mathcal{T}\cup\{\mathcal{A}(k)\}}(u_{\mathcal{L}_{\mathcal{T}}^{-1}(i)})$.

In addition, the combination k' and $\mathbf{u}' = (u'_1, \ldots, u'_{|\mathcal{L}_{\mathcal{T}}|})$ also causes a term in (82) which has the product

$$x_{\mathcal{A}(k'),i} \prod_{i_2\in[|\mathcal{L}_{\mathcal{T}}|]} x_{\mathcal{A}\left(\overline{\text{Ind}}_{\mathcal{T}\cup\{\mathcal{A}(k')\}}(u'_{i_2})\right),\mathcal{L}_{\mathcal{T}}(i_2)}. \tag{86}$$

The products in (84) and (86) are identical if $\mathbf{u}'$ is as follows,

- for $j \in [|\mathcal{L}_{\mathcal{T}}|]\setminus\{\mathcal{L}_{\mathcal{T}}^{-1}(i)\}$, we have

$$\mathcal{A}\left(\overline{\text{Ind}}_{\mathcal{T}\cup\{\mathcal{A}(k')\}}(u'_j)\right) = \mathcal{A}\left(\overline{\text{Ind}}_{\mathcal{T}\cup\{\mathcal{A}(k)\}}(u_j)\right); \tag{87}$$

  such that

$$x_{\mathcal{A}\left(\overline{\text{Ind}}_{\mathcal{T}\cup\{\mathcal{A}(k')\}}(u'_j)\right),\mathcal{L}_{\mathcal{T}}(j)} = x_{\mathcal{A}\left(\overline{\text{Ind}}_{\mathcal{T}\cup\{\mathcal{A}(k)\}}(u_j)\right),\mathcal{L}_{\mathcal{T}}(j)}; \tag{88}$$

- for $j = \mathcal{L}_{\mathcal{T}}^{-1}(i)$, we have

$$\mathcal{A}\left(\overline{\text{Ind}}_{\mathcal{T}\cup\{\mathcal{A}(k')\}}(u'_j)\right) = \mathcal{A}(k); \tag{89}$$

  such that

$$x_{\mathcal{A}\left(\overline{\text{Ind}}_{\mathcal{T}\cup\{\mathcal{A}(k')\}}(u'_j)\right),\mathcal{L}_{\mathcal{T}}(j)} = x_{\mathcal{A}(k),i}. \tag{90}$$

It is obvious to check that there does not exist any other combination of $k'' \in \overline{\text{Ind}}_{\mathcal{T}}$ and $\mathbf{u}'' \in \text{Perm}(|\mathcal{L}_{\mathcal{T}}|)$, causing a term on the LHS of (82) which has the product in (84), except the two above combinations.

In Appendix C, we will prove that

$$(-1)^{-\overline{\text{Ind}}_{\mathcal{T}}^{-1}(k)+1+\text{Tot}(\overline{\text{Ind}}_{\mathcal{T}})+\sum_{i_1\in[|\mathcal{L}_{\mathcal{T}}|]}\text{Card}([|\mathcal{L}_{\mathcal{T}}|]\setminus\{u_1,\ldots,u_{i_1}\},u_{i_1})} + (-1)^{-\overline{\text{Ind}}_{\mathcal{T}}^{-1}(k')+1+\text{Tot}(\overline{\text{Ind}}_{\mathcal{T}})+\sum_{i'_1\in[|\mathcal{L}_{\mathcal{T}}|]}\text{Card}([|\mathcal{L}_{\mathcal{T}}|]\setminus\{u'_1,\ldots,u'_{i'_1}\},u'_{i'_1})} = 0, \tag{91}$$

such that the coefficient of the product in (84) on the LHS of (82) is 0. In other words, for each combination of k and $\mathbf{u}$ on the LHS of (82), there is exactly one term caused by the combination of k' and $\mathbf{u}'$, such that the sum of these two caused terms is 0. Thus (82) is proved.

Remark 9 (Connection to the Example in Section IV-A – Part IV): In the illustrated example, let us focus on $W_{1,\{1\}}$ which is in Case 2. It can be seen that $\overline{\text{Ind}}_{\{1\}} = [2]$. For each $k \in [2]$, $W_{1,\{1\}}$ appears in $X_{\{1\}\cup\mathcal{A}(k)}$. When $k = 1$, we have

$$X_{\{1\}\cup\mathcal{A}(k)} = X_{\{1,3\}} = W_{1,\{3\}} + (y_{3,1}W_{1,\{1\}} + y_{3,2}W_{2,\{1\}}). \tag{92}$$

From (71), the coefficient of $X_{\{1,3\}}$ in (74) is (recall that $\mathcal{L}_{\{1,3\}} = \{1\}$ and $\overline{\text{Ind}}_{\{1,3\}} = \{2\}$)

$$\beta_{\{3,4\},\{1,3\}} = \sum_{\mathbf{u}\in\text{Perm}(1)} (-1)^{1+2+\text{Card}([1]\setminus\{u_1\},u_1)} y_{4,1} = -y_{4,1}. \tag{93}$$

$$\beta_{\mathcal{A},\mathcal{S}} = \sum_{\substack{\mathbf{u}=(u_1,\ldots,u_{|\mathcal{L}_{\mathcal{S}}|})\\ \in\text{Perm}(|\mathcal{L}_{\mathcal{S}}|)}} (-1)^{1+\text{Tot}(\overline{\text{Ind}}_{\mathcal{S}})+\sum_{i_1\in[|\mathcal{L}_{\mathcal{S}}|]}\text{Card}([|\mathcal{L}_{\mathcal{S}}|]\setminus\{u_1,\ldots,u_{i_1}\},u_{i_1})} \prod_{i_2\in[|\mathcal{L}_{\mathcal{S}}|]} x_{\mathcal{A}\left(\overline{\text{Ind}}_{\mathcal{S}}(u_{i_2})\right),\mathcal{L}_{\mathcal{S}}(i_2)}. \tag{71}$$

When $k = 2$, we have

$$X_{\{1\}\cup\mathcal{A}(k)} = X_{\{1,4\}} = W_{1,\{4\}} + (y_{4,1}W_{1,\{1\}} + y_{4,2}W_{2,\{1\}}). \tag{94}$$

From (71), the coefficient of $X_{\{1,4\}}$ in (74) is (recall that $\mathcal{L}_{\{1,4\}} = \{1\}$ and $\overline{\text{Ind}}_{\{1,4\}} = \{1\}$)

$$\beta_{\{3,4\},\{1,4\}} = \sum_{\mathbf{u}\in\text{Perm}(1)} (-1)^{1+1+\text{Card}([1]\setminus\{u_1\},u_1)} y_{3,1} = y_{3,1}. \tag{95}$$

Hence, the coefficient of $W_{1,\{1\}}$ in (74) is

$$y_{3,1}\beta_{\{3,4\},\{1,3\}} + y_{4,1}\beta_{\{3,4\},\{1,4\}} = y_{3,1}(-y_{4,1}) + y_{4,1}y_{3,1} = 0. \tag{96}$$

□

C. Case 3

Lastly we focus on one leader block $B_{i,\mathcal{T}}$ in Hierarchy $h > 0$ where $i \notin \mathcal{T}$. By definition, we have $|\mathcal{T} \cap \mathcal{L}| = h$. On the RHS of (70), as $i \notin \mathcal{T}$, $B_{i,\mathcal{T}}$ appears in $X_{\mathcal{T}\cup\{i\}}$. In addition, $B_{i,\mathcal{T}}$ also appears in $X_{\mathcal{T}\cup\{\mathcal{A}(k)\}}$, where $k \in \overline{\text{Ind}}_{\mathcal{T}}$.

Let us first focus on $X_{\mathcal{T}\cup\{i\}}$. Recall that the $\left(\mathcal{L}^{-1}_{\mathcal{T}\cup\{i\}}(i)\right)^{\text{th}}$ element in $\mathcal{L}_{\mathcal{T}\cup\{i\}}$ is i. From (54), it can be seen that the coefficient of $B_{i,\mathcal{T}}$ in $X_{\mathcal{T}\cup\{i\}}$ is

$$(-1)^{\mathcal{L}^{-1}_{\mathcal{T}\cup\{i\}}(i)-1}. \tag{97}$$

In addition, we have (98), shown at the bottom of the next page.

Let us then focus on $X_{\mathcal{T}\cup\{\mathcal{A}(k)\}}$, where $k \in \overline{\text{Ind}}_{\mathcal{T}}$. It was proved in (80) that the coefficient of $B_{i,\mathcal{T}}$ in $X_{\mathcal{T}\cup\{\mathcal{A}(k)\}}$ is

$$(-1)^{k-\overline{\text{Ind}}^{-1}_{\mathcal{T}}(k)} x_{\mathcal{A}(k),i}. \tag{99}$$

In addition, it was proved in (81b) that

$$\beta_{\mathcal{A},\mathcal{T}\cup\{\mathcal{A}(k)\}} = \sum_{\substack{\mathbf{u}=(u_1,\ldots,u_{|\mathcal{L}_{\mathcal{T}}|})\\ \in\text{Perm}(|\mathcal{L}_{\mathcal{T}}|)}} (-1)^{1+\left(\text{Tot}(\overline{\text{Ind}}_{\mathcal{T}})-k\right)+\sum_{i_1\in[|\mathcal{L}_{\mathcal{T}}|]}\text{Card}([|\mathcal{L}_{\mathcal{T}}|]\setminus\{u_1,\ldots,u_{i_1}\},u_{i_1})} \prod_{i_2\in[|\mathcal{L}_{\mathcal{T}}|]} x_{\mathcal{A}\left(\overline{\text{Ind}}_{\mathcal{T}\cup\{\mathcal{A}(k)\}}(u_{i_2})\right),\mathcal{L}_{\mathcal{T}}(i_2)}. \tag{100}$$

In the following, we will prove

$$(-1)^{\mathcal{L}^{-1}_{\mathcal{T}\cup\{i\}}(i)-1}\beta_{\mathcal{A},\mathcal{T}\cup\{i\}} + \sum_{k\in\overline{\text{Ind}}_{\mathcal{T}}} (-1)^{k-\overline{\text{Ind}}^{-1}_{\mathcal{T}}(k)} x_{\mathcal{A}(k),i}\beta_{\mathcal{A},\mathcal{T}\cup\{\mathcal{A}(k)\}} = 0, \tag{101}$$

such that the coefficient of $B_{i,\mathcal{T}}$ on the RHS of (70) is 0. Note that there are $t - |\mathcal{L}_{\mathcal{T}}|$ non-leaders in $\mathcal{T}$. As there are totally $t + 1$ non-leaders in $\mathcal{A}$, we have

$$|\overline{\text{Ind}}_{\mathcal{T}}| = t + 1 - (t - |\mathcal{L}_{\mathcal{T}}|) = |\mathcal{L}_{\mathcal{T}}| + 1. \tag{102}$$

Let us focus on one permutation $\mathbf{u} = (u_1, \ldots, u_{|\mathcal{L}_{\mathcal{T}}|+1}) \in \text{Perm}(|\mathcal{L}_{\mathcal{T}}| + 1)$ in $\beta_{\mathcal{A},\mathcal{T}\cup\{i\}}$. The term in (101) caused by $\mathbf{u}$ is (103), shown at the bottom of the next page.

We can rewrite the product term in (103b) as follows (recall again that the $\left(\mathcal{L}^{-1}_{\mathcal{T}\cup\{i\}}(i)\right)^{\text{th}}$ element in $\mathcal{L}_{\mathcal{T}\cup\{i\}}$ is i),

$$\prod_{i_2\in[|\mathcal{L}_{\mathcal{T}}|+1]} x_{\mathcal{A}\left(\overline{\text{Ind}}_{\mathcal{T}}(u_{i_2})\right),\mathcal{L}_{\mathcal{T}\cup\{i\}}(i_2)} = x_{\mathcal{A}\left(\overline{\text{Ind}}_{\mathcal{T}}(u_{\mathcal{L}^{-1}_{\mathcal{T}\cup\{i\}}(i)})\right),i} \prod_{\substack{i_2\in[|\mathcal{L}_{\mathcal{T}}|+1]\\ \setminus\{\mathcal{L}^{-1}_{\mathcal{T}\cup\{i\}}(i)\}}} x_{\mathcal{A}\left(\overline{\text{Ind}}_{\mathcal{T}}(u_{i_2})\right),\mathcal{L}_{\mathcal{T}\cup\{i\}}(i_2)} \tag{104a}$$

$$= x_{\mathcal{A}(\widetilde{k}),i} \prod_{i_2\in[|\mathcal{L}_{\mathcal{T}}|+1]\setminus\{\mathcal{L}^{-1}_{\mathcal{T}\cup\{i\}}(i)\}} x_{\mathcal{A}\left(\overline{\text{Ind}}_{\mathcal{T}}(u_{i_2})\right),\mathcal{L}_{\mathcal{T}\cup\{i\}}(i_2)}, \tag{104b}$$

$$\beta_{\mathcal{A},\mathcal{T}\cup\{\mathcal{A}(k)\}} = \sum_{\substack{\mathbf{u}=(u_1,\ldots,u_{|\mathcal{L}_{\mathcal{T}\cup\{\mathcal{A}(k)\}}|})\\ \in\text{Perm}(|\mathcal{L}_{\mathcal{T}\cup\{\mathcal{A}(k)\}}|)}} (-1)^{1+\text{Tot}(\overline{\text{Ind}}_{\mathcal{T}\cup\{\mathcal{A}(k)\}})+\sum_{i_1\in[|\mathcal{L}_{\mathcal{T}\cup\{\mathcal{A}(k)\}}|]}\text{Card}([|\mathcal{L}_{\mathcal{T}\cup\{\mathcal{A}(k)\}}|]\setminus\{u_1,\ldots,u_{i_1}\},u_{i_1})} \prod_{i_2\in[|\mathcal{L}_{\mathcal{T}\cup\{\mathcal{A}(k)\}}|]} x_{\mathcal{A}\left(\overline{\text{Ind}}_{\mathcal{T}\cup\{\mathcal{A}(k)\}}(u_{i_2})\right),\mathcal{L}_{\mathcal{T}\cup\{\mathcal{A}(k)\}}(i_2)} \tag{81a}$$

$$= \sum_{\substack{\mathbf{u}=(u_1,\ldots,u_{|\mathcal{L}_{\mathcal{T}}|})\\ \in\text{Perm}(|\mathcal{L}_{\mathcal{T}}|)}} (-1)^{1+\left(\text{Tot}(\overline{\text{Ind}}_{\mathcal{T}})-k\right)+\sum_{i_1\in[|\mathcal{L}_{\mathcal{T}}|]}\text{Card}([|\mathcal{L}_{\mathcal{T}}|]\setminus\{u_1,\ldots,u_{i_1}\},u_{i_1})} \prod_{i_2\in[|\mathcal{L}_{\mathcal{T}}|]} x_{\mathcal{A}\left(\overline{\text{Ind}}_{\mathcal{T}\cup\{\mathcal{A}(k)\}}(u_{i_2})\right),\mathcal{L}_{\mathcal{T}}(i_2)}. \tag{81b}$$

$$(-1)^{k-\overline{\text{Ind}}^{-1}_{\mathcal{T}}(k)} x_{\mathcal{A}(k),i} \left\{(-1)^{1+\left(\text{Tot}(\overline{\text{Ind}}_{\mathcal{T}})-k\right)+\sum_{i_1\in[|\mathcal{L}_{\mathcal{T}}|]}\text{Card}([|\mathcal{L}_{\mathcal{T}}|]\setminus\{u_1,\ldots,u_{i_1}\},u_{i_1})} \prod_{i_2\in[|\mathcal{L}_{\mathcal{T}}|]} x_{\mathcal{A}\left(\overline{\text{Ind}}_{\mathcal{T}\cup\{\mathcal{A}(k)\}}(u_{i_2})\right),\mathcal{L}_{\mathcal{T}}(i_2)}\right\} \tag{83a}$$

$$= (-1)^{-\overline{\text{Ind}}^{-1}_{\mathcal{T}}(k)+1+\text{Tot}(\overline{\text{Ind}}_{\mathcal{T}})+\sum_{i_1\in[|\mathcal{L}_{\mathcal{T}}|]}\text{Card}([|\mathcal{L}_{\mathcal{T}}|]\setminus\{u_1,\ldots,u_{i_1}\},u_{i_1})} \left\{x_{\mathcal{A}(k),i} \prod_{i_2\in[|\mathcal{L}_{\mathcal{T}}|]} x_{\mathcal{A}\left(\overline{\text{Ind}}_{\mathcal{T}\cup\{\mathcal{A}(k)\}}(u_{i_2})\right),\mathcal{L}_{\mathcal{T}}(i_2)}\right\}. \tag{83b}$$

where we define $\widetilde{k} := \overline{\text{Ind}}_{\mathcal{T}}(u_{\mathcal{L}^{-1}_{\mathcal{T}\cup\{i\}}(i)})$.

Hence, on the LHS of (101), besides $(-1)^{\mathcal{L}^{-1}_{\mathcal{T}\cup\{i\}}(i)-1}\beta_{\mathcal{A},\mathcal{T}\cup\{i\}}$, only the caused term by the combination of $\widetilde{k}$ and $\widetilde{\mathbf{u}} = (\widetilde{u}_1, \ldots, \widetilde{u}_{|\mathcal{L}_{\mathcal{T}}|})$ has the product in (104b), where

$$\widetilde{\mathbf{u}} = \Big(g(u_1), \ldots, g(u_{\mathcal{L}^{-1}_{\mathcal{T}\cup\{i\}}(i)-1}), g(u_{\mathcal{L}^{-1}_{\mathcal{T}\cup\{i\}}(i)+1}), \ldots, g(u_{|\mathcal{L}_{\mathcal{T}}|+1})\Big), \tag{105a}$$

$$g(u_j) := \begin{cases} u_j, & \text{if } u_j < u_{\mathcal{L}^{-1}_{\mathcal{T}\cup\{i\}}(i)} \\ u_j - 1 & \text{if } u_j > u_{\mathcal{L}^{-1}_{\mathcal{T}\cup\{i\}}(i)} \end{cases}. \tag{105b}$$

In Appendix D, we will prove that

$$(-1)^{\mathcal{L}^{-1}_{\mathcal{T}\cup\{i\}}(i)+\text{Tot}(\overline{\text{Ind}}_{\mathcal{T}})+\sum_{i_1\in[|\mathcal{L}_{\mathcal{T}}|+1]}\text{Card}([|\mathcal{L}_{\mathcal{T}}|+1]\setminus\{u_1,\ldots,u_{i_1}\},u_{i_1})} + (-1)^{-\overline{\text{Ind}}^{-1}_{\mathcal{T}}(\widetilde{k})+1+\text{Tot}(\overline{\text{Ind}}_{\mathcal{T}})+\sum_{\widetilde{i}_1\in[|\mathcal{L}_{\mathcal{T}}|]}\text{Card}([|\mathcal{L}_{\mathcal{T}}|]\setminus\{\widetilde{u}_1,\ldots,\widetilde{u}_{\widetilde{i}_1}\},\widetilde{u}_{\widetilde{i}_1})} = 0, \tag{106}$$

such that the coefficient of the product in (104b) on the LHS of (101) is 0. Hence, for each permutation $\mathbf{u} \in \text{Perm}(|\mathcal{L}_{\mathcal{T}}|+1)$, there is exactly one term caused by the combination of $\widetilde{k} \in \overline{\text{Ind}}_{\mathcal{T}}$ and $\widetilde{\mathbf{u}} \in \text{Perm}(|\mathcal{L}_{\mathcal{T}}|)$, such that the sum of these two caused terms are 0.

In addition, on the LHS of (101), there are $(|\mathcal{L}_{\mathcal{T}}|+1)!$ terms in $(-1)^{\mathcal{L}^{-1}_{\mathcal{T}\cup\{i\}}(i)-1}\beta_{\mathcal{A},\mathcal{T}\cup\{i\}}$. Recall that in (102), we proved $|\overline{\text{Ind}}_{\mathcal{T}}| = |\mathcal{L}_{\mathcal{T}}|+1$. Hence, on the LHS of (101), there are $|\mathcal{L}_{\mathcal{T}}|! \times (|\mathcal{L}_{\mathcal{T}}|+1) = (|\mathcal{L}_{\mathcal{T}}|+1)!$ terms in $\sum_{k\in\overline{\text{Ind}}_{\mathcal{T}}}(-1)^{k-\overline{\text{Ind}}^{-1}_{\mathcal{T}}(k)}x_{\mathcal{A}(k),i}\beta_{\mathcal{A},\mathcal{T}\cup\{\mathcal{A}(k)\}}$. In conclusion, we prove (101).

Remark 10 (Connection to the Example in Section IV-A – Part V): In the illustrated example, let us focus on $W_{1,\{2\}}$ which is in Case 3. It can be seen that $W_{1,\{2\}}$ appears in $X_{\{1,2\}}$, where

$$X_{\{1,2\}} = W_{1,\{2\}} - W_{2,\{1\}}. \tag{107}$$

From (71), the coefficient of $X_{\{1,2\}}$ in (74) is (recall that $\mathcal{L}_{\{1,2\}} = [2]$ and $\overline{\text{Ind}}_{\{1,2\}} = [2]$)

$$\beta_{\{3,4\},\{1,2\}} = \sum_{\mathbf{u}\in\text{Perm}(2)}(-1)^{1+3+\text{Card}([2]\setminus\{u_1\},u_1)}\prod_{i_2\in[2]}y_{\mathcal{A}(u_{i_2}),i_2} \tag{108a}$$

$$= y_{3,1}y_{4,2} - y_{4,1}y_{3,2}. \tag{108b}$$

In addition, with $\overline{\text{Ind}}_{\{2\}} = [2]$, it can be seen that $W_{1,\{2\}}$ also appears in $X_{\{2\}\cup\mathcal{A}(k)}$ where $k \in [2]$. When $k = 1$, we have

$$X_{\{2\}\cup\mathcal{A}(k)} = X_{\{2,3\}} = W_{2,\{3\}} + (y_{3,1}W_{1,\{2\}} + y_{3,2}W_{2,\{2\}}). \tag{109}$$

From (71), the coefficient of $X_{\{2,3\}}$ in (74) is (recall that $\mathcal{L}_{\{2,3\}} = \{2\}$ and $\overline{\text{Ind}}_{\{2,3\}}) = \{2\}$)

$$\beta_{\{3,4\},\{2,3\}} = \sum_{\mathbf{u}\in\text{Perm}(1)}(-1)^{1+2}y_{4,2} = -y_{4,2}. \tag{110}$$

When $k = 2$, we have

$$X_{\{2\}\cup\mathcal{A}(k)} = X_{\{2,4\}} = W_{2,\{4\}} + (y_{4,1}W_{1,\{2\}} + y_{4,2}W_{2,\{2\}}). \tag{111}$$

From (71), the coefficient of $X_{\{2,4\}}$ in (74) is (recall that $\mathcal{L}_{\{2,4\}} = \{2\}$ and $\overline{\text{Ind}}_{\{2,4\}}) = \{1\}$)

$$\beta_{\{3,4\},\{2,4\}} = \sum_{\mathbf{u}\in\text{Perm}(1)}(-1)^{1+1}y_{3,2} = y_{3,2}. \tag{112}$$

Hence, the coefficient of $W_{1,\{2\}}$ in (74) is

$$\beta_{\{3,4\},\{1,2\}} + y_{3,1}\beta_{\{3,4\},\{2,3\}} + y_{4,1}\beta_{\{3,4\},\{2,4\}} \tag{113a}$$

$$= y_{3,1}y_{4,2} - y_{4,1}y_{3,2} + y_{3,1}(-y_{4,2}) + y_{4,1}y_{3,2} = 0. \tag{113b}$$

□

$$\beta_{\mathcal{A},\mathcal{T}\cup\{i\}} = \sum_{\substack{\mathbf{u}=(u_1,\ldots,u_{|\mathcal{L}_{\mathcal{T}\cup\{i\}}|})\\ \in\text{Perm}(|\mathcal{L}_{\mathcal{T}\cup\{i\}}|)}}(-1)^{1+\text{Tot}(\overline{\text{Ind}}_{\mathcal{T}\cup\{i\}})+\sum_{i_1\in[|\mathcal{L}_{\mathcal{T}\cup\{i\}}|]}\text{Card}([|\mathcal{L}_{\mathcal{T}\cup\{i\}}|]\setminus\{u_1,\ldots,u_{i_1}\},u_{i_1})}\prod_{i_2\in[|\mathcal{L}_{\mathcal{T}\cup\{i\}}|]}x_{\mathcal{A}\left(\overline{\text{Ind}}_{\mathcal{T}\cup\{i\}}(u_{i_2})\right),\mathcal{L}_{\mathcal{T}\cup\{i\}}(i_2)} \tag{98a}$$

$$= \sum_{\substack{\mathbf{u}=(u_1,\ldots,u_{|\mathcal{L}_{\mathcal{T}}|+1})\\ \in\text{Perm}(|\mathcal{L}_{\mathcal{T}}|+1)}}(-1)^{1+\text{Tot}(\overline{\text{Ind}}_{\mathcal{T}})+\sum_{i_1\in[|\mathcal{L}_{\mathcal{T}}|+1]}\text{Card}([|\mathcal{L}_{\mathcal{T}}|+1]\setminus\{u_1,\ldots,u_{i_1}\},u_{i_1})}\prod_{i_2\in[|\mathcal{L}_{\mathcal{T}}|+1]}x_{\mathcal{A}\left(\overline{\text{Ind}}_{\mathcal{T}}(u_{i_2})\right),\mathcal{L}_{\mathcal{T}\cup\{i\}}(i_2)} \tag{98b}$$

$$(-1)^{\mathcal{L}^{-1}_{\mathcal{T}\cup\{i\}}(i)-1}\left\{(-1)^{1+\text{Tot}(\overline{\text{Ind}}_{\mathcal{T}})+\sum_{i_1\in[|\mathcal{L}_{\mathcal{T}}|+1]}\text{Card}([|\mathcal{L}_{\mathcal{T}}|+1]\setminus\{u_1,\ldots,u_{i_1}\},u_{i_1})}\prod_{i_2\in[|\mathcal{L}_{\mathcal{T}}|+1]}x_{\mathcal{A}\left(\overline{\text{Ind}}_{\mathcal{T}}(u_{i_2})\right),\mathcal{L}_{\mathcal{T}\cup\{i\}}(i_2)}\right\} \tag{103a}$$

$$= (-1)^{\mathcal{L}^{-1}_{\mathcal{T}\cup\{i\}}(i)+\text{Tot}(\overline{\text{Ind}}_{\mathcal{T}})+\sum_{i_1\in[|\mathcal{L}_{\mathcal{T}}|+1]}\text{Card}([|\mathcal{L}_{\mathcal{T}}|+1]\setminus\{u_1,\ldots,u_{i_1}\},u_{i_1})}\left\{\prod_{i_2\in[|\mathcal{L}_{\mathcal{T}}|+1]}x_{\mathcal{A}\left(\overline{\text{Ind}}_{\mathcal{T}}(u_{i_2})\right),\mathcal{L}_{\mathcal{T}\cup\{i\}}(i_2)}\right\}. \tag{103b}$$

APPENDIX C
PROOF OF (91)

To prove (91), it is equivalent to prove, (114), shown at the bottom of the page.

Let us focus on $\sum_{i_1\in[|\mathcal{L}_{\mathcal{T}}|]} \mathrm{Card}([|\mathcal{L}_{\mathcal{T}}|]\setminus\{u_1,\ldots,u_{i_1}\},u_{i_1})$. By the definition of the function $\mathrm{Card}(\cdot)$ in (69), we have

$$\begin{aligned}
&\sum_{i_1\in[|\mathcal{L}_{\mathcal{T}}|]} \mathrm{Card}([|\mathcal{L}_{\mathcal{T}}|]\setminus\{u_1,\ldots,u_{i_1}\},u_{i_1})\\
&= \sum_{i_1\in[|\mathcal{L}_{\mathcal{T}}|]:i_1\neq\mathcal{L}_{\mathcal{T}}^{-1}(i)} \mathrm{Card}\big(([|\mathcal{L}_{\mathcal{T}}|]\setminus\{u_{\mathcal{L}_{\mathcal{T}}^{-1}(i)}\})\\
&\setminus\{u_1,\ldots,u_{\mathcal{L}_{\mathcal{T}}^{-1}(i)-1},u_{\mathcal{L}_{\mathcal{T}}^{-1}(i)+1},\ldots,u_{i_1}\},u_{i_1}\big)\\
&+|\{i_2\in[\mathcal{L}_{\mathcal{T}}^{-1}(i)-1]:u_{\mathcal{L}_{\mathcal{T}}^{-1}(i)}<u_{i_2}\}|\\
&+|\{i_3\in[\mathcal{L}_{\mathcal{T}}^{-1}(i)+1:|\mathcal{L}_{\mathcal{T}}|]:u_{i_3}<u_{\mathcal{L}_{\mathcal{T}}^{-1}(i)}\}| \qquad (115a)\\
&= \sum_{i_1\in[|\mathcal{L}_{\mathcal{T}}|]:i_1\neq\mathcal{L}_{\mathcal{T}}^{-1}(i)} \mathrm{Card}\big(([|\mathcal{L}_{\mathcal{T}}|]\\
&\setminus\{u_{\mathcal{L}_{\mathcal{T}}^{-1}(i)}\})\setminus\{u_1,\ldots,u_{\mathcal{L}_{\mathcal{T}}^{-1}(i)-1},u_{\mathcal{L}_{\mathcal{T}}^{-1}(i)+1},\ldots,u_{i_1}\},u_{i_1}\big)\\
&+(\mathcal{L}_{\mathcal{T}}^{-1}(i)-1-|\{i_2\in[\mathcal{L}_{\mathcal{T}}^{-1}(i)-1]:u_{i_2}<u_{\mathcal{L}_{\mathcal{T}}^{-1}(i)}\}|)\\
&+|\{i_3\in[\mathcal{L}_{\mathcal{T}}^{-1}(i)+1:|\mathcal{L}_{\mathcal{T}}|]:u_{i_3}<u_{\mathcal{L}_{\mathcal{T}}^{-1}(i)}\}| \qquad (115b)\\
&= \sum_{i_1\in[|\mathcal{L}_{\mathcal{T}}|]:i_1\neq\mathcal{L}_{\mathcal{T}}^{-1}(i)} \mathrm{Card}\big(([|\mathcal{L}_{\mathcal{T}}|]\\
&\setminus\{u_{\mathcal{L}_{\mathcal{T}}^{-1}(i)}\})\setminus\{u_1,\ldots,u_{\mathcal{L}_{\mathcal{T}}^{-1}(i)-1},u_{\mathcal{L}_{\mathcal{T}}^{-1}(i)+1},\ldots,u_{i_1}\},u_{i_1}\big)\\
&+(\mathcal{L}_{\mathcal{T}}^{-1}(i)-1-|\{i_2\in[\mathcal{L}_{\mathcal{T}}^{-1}(i)-1]:u_{i_2}<u_{\mathcal{L}_{\mathcal{T}}^{-1}(i)}\}|)\\
&+\mathrm{Card}([|\mathcal{L}_{\mathcal{T}}|],u_{\mathcal{L}_{\mathcal{T}}^{-1}(i)})-|\{i_2\in[\mathcal{L}_{\mathcal{T}}^{-1}(i)-1]:u_{i_2}<u_{\mathcal{L}_{\mathcal{T}}^{-1}(i)}\}|. \qquad (115c)
\end{aligned}$$

Similarly, for $\sum_{i'_1\in[|\mathcal{L}_{\mathcal{T}}|]} \mathrm{Card}([|\mathcal{L}_{\mathcal{T}}|]\setminus\{u'_1,\ldots,u'_{i'_1}\},u'_{i'_1})$, from the same derivation as (115c), we have

$$\begin{aligned}
&\sum_{i'_1\in[|\mathcal{L}_{\mathcal{T}}|]} \mathrm{Card}([|\mathcal{L}_{\mathcal{T}}|]\setminus\{u'_1,\ldots,u'_{i'_1}\},u'_{i'_1})\\
&= \sum_{i'_1\in[|\mathcal{L}_{\mathcal{T}}|]:i'_1\neq\mathcal{L}_{\mathcal{T}}^{-1}(i)} \mathrm{Card}\big(([|\mathcal{L}_{\mathcal{T}}|]\\
&\setminus\{u'_{\mathcal{L}_{\mathcal{T}}^{-1}(i)}\})\setminus\{u'_1,\ldots,u'_{\mathcal{L}_{\mathcal{T}}^{-1}(i)-1},u'_{\mathcal{L}_{\mathcal{T}}^{-1}(i)+1},\ldots,u'_{i_1}\},u'_{i_1}\big)\\
&+(\mathcal{L}_{\mathcal{T}}^{-1}(i)-1-|\{i'_2\in[\mathcal{L}_{\mathcal{T}}^{-1}(i)-1]:u'_{i'_2}<u'_{\mathcal{L}_{\mathcal{T}}^{-1}(i)}\}|)\\
&+\mathrm{Card}([|\mathcal{L}_{\mathcal{T}}|],u'_{\mathcal{L}_{\mathcal{T}}^{-1}(i)})-|\{i'_2\in[\mathcal{L}_{\mathcal{T}}^{-1}(i)-1]:u'_{i'_2}<u'_{\mathcal{L}_{\mathcal{T}}^{-1}(i)}\}|. \qquad (116)
\end{aligned}$$

In addition, from (87), it can be seen that

$$\begin{aligned}
&\sum_{i_1\in[|\mathcal{L}_{\mathcal{T}}|]:i_1\neq\mathcal{L}_{\mathcal{T}}^{-1}(i)} \mathrm{Card}\big(([|\mathcal{L}_{\mathcal{T}}|]\setminus\{u_{\mathcal{L}_{\mathcal{T}}^{-1}(i)}\})\\
&\setminus\{u_1,\ldots,u_{\mathcal{L}_{\mathcal{T}}^{-1}(i)-1},u_{\mathcal{L}_{\mathcal{T}}^{-1}(i)+1},\ldots,u_{i_1}\},u_{i_1}\big)\\
&= \sum_{i'_1\in[|\mathcal{L}_{\mathcal{T}}|]:i'_1\neq\mathcal{L}_{\mathcal{T}}^{-1}(i)} \mathrm{Card}\big(([|\mathcal{L}_{\mathcal{T}}|]\\
&\setminus\{u'_{\mathcal{L}_{\mathcal{T}}^{-1}(i)}\})\setminus\{u'_1,\ldots,u'_{\mathcal{L}_{\mathcal{T}}^{-1}(i)-1},u'_{\mathcal{L}_{\mathcal{T}}^{-1}(i)+1},\ldots,u'_{i_1}\},u'_{i_1}\big). \qquad (117)
\end{aligned}$$

From (115c)-(117), and the fact that $(-1)^{2a}=(-1)^0$ for any integer a, we have, (118), shown at the top of the next page.

Without loss of generality, we assume $k<k'$. Recall that $\overline{\mathrm{Ind}}_{\mathcal{T}\cup\{\mathcal{A}(k)\}}(u_{\mathcal{L}_{\mathcal{T}}^{-1}(i)})=k'$. By the definition in (79), we can see that in $\overline{\mathrm{Ind}}_{\mathcal{T}}$, there are $\overline{\mathrm{Ind}}_{\mathcal{T}}^{-1}(k')-1$ elements smaller than k'. By the assumption, $k<k'$. Hence, in $\overline{\mathrm{Ind}}_{\mathcal{T}\cup\{\mathcal{A}(k)\}}$, there are $\overline{\mathrm{Ind}}_{\mathcal{T}}^{-1}(k')-2$ elements smaller than k'. In other words,

$$u_{\mathcal{L}_{\mathcal{T}}^{-1}(i)}=\overline{\mathrm{Ind}}_{\mathcal{T}}^{-1}(k')-1, \qquad (119)$$

which leads to

$$\mathrm{Card}(\{u_1,\ldots,u_{|\mathcal{L}_{\mathcal{T}}|}\},u_{\mathcal{L}_{\mathcal{T}}^{-1}(i)})=\overline{\mathrm{Ind}}_{\mathcal{T}}^{-1}(k')-2. \qquad (120)$$

Similarly, recall that $\overline{\mathrm{Ind}}_{\mathcal{T}\cup\{\mathcal{A}(k')\}}(u'_{\mathcal{L}_{\mathcal{T}}^{-1}(i)})=k$. In $\overline{\mathrm{Ind}}_{\mathcal{T}}$, there are $\overline{\mathrm{Ind}}_{\mathcal{T}}^{-1}(k)-1$ elements smaller than k. By the assumption, $k<k'$. Hence, in $\overline{\mathrm{Ind}}_{\mathcal{T}\cup\{\mathcal{A}(k')\}}$, there are $\overline{\mathrm{Ind}}_{\mathcal{T}}^{-1}(k)-1$ elements smaller than k. In other words,

$$u'_{\mathcal{L}_{\mathcal{T}}^{-1}(i)}=\overline{\mathrm{Ind}}_{\mathcal{T}}^{-1}(k), \qquad (121)$$

which leads to

$$\mathrm{Card}(\{u'_1,\ldots,u'_{|\mathcal{L}_{\mathcal{T}}|}\},u'_{\mathcal{L}_{\mathcal{T}}^{-1}(i)})=\overline{\mathrm{Ind}}_{\mathcal{T}}^{-1}(k)-1. \qquad (122)$$

We take (120) and (122) into (118) to obtain, (123), shown at the top of the next page.

Finally, we have, (124), shown at the top of the next page. which proves (114).

APPENDIX D
PROOF OF (106)

To prove (106), it is equivalent to prove, (125), shown at the top of the next page.

Let us focus on $\sum_{i_1\in[|\mathcal{L}_{\mathcal{T}}|+1]} \mathrm{Card}([|\mathcal{L}_{\mathcal{T}}|+1]\setminus\{u_1,\ldots,u_{i_1}\},u_{i_1})$. By the definition of the function $\mathrm{Card}(\cdot)$ in (69), we have

$$\begin{aligned}
&\sum_{i_1\in[|\mathcal{L}_{\mathcal{T}}|+1]} \mathrm{Card}([|\mathcal{L}_{\mathcal{T}}|+1]\setminus\{u_1,\ldots,u_{i_1}\},u_{i_1})\\
&= \sum_{\substack{i_1\in[|\mathcal{L}_{\mathcal{T}}|+1]:\\ i_1\neq\mathcal{L}_{\mathcal{T}\cup\{i\}}^{-1}(i)}} \mathrm{Card}\big(([|\mathcal{L}_{\mathcal{T}}|+1]\setminus\{u_{\mathcal{L}_{\mathcal{T}\cup\{i\}}^{-1}(i)}\})\\
&\setminus\{u_1,\ldots,u_{\mathcal{L}_{\mathcal{T}\cup\{i\}}^{-1}(i)-1},u_{\mathcal{L}_{\mathcal{T}\cup\{i\}}^{-1}(i)+1},\ldots,u_{i_1}\},u_{i_1}\big)\\
&+|\{i_2\in[\mathcal{L}_{\mathcal{T}\cup\{i\}}^{-1}(i)-1]:u_{\mathcal{L}_{\mathcal{T}\cup\{i\}}^{-1}(i)}<u_{i_2}\}|\\
&+|\{i_3\in[\mathcal{L}_{\mathcal{T}\cup\{i\}}^{-1}(i)+1:|\mathcal{L}_{\mathcal{T}}|+1]:u_{i_3}<u_{|\mathcal{L}_{\mathcal{T}}|+1}\}| \qquad (126a)\\
&= \sum_{\substack{i_1\in[|\mathcal{L}_{\mathcal{T}}|+1]:\\ i_1\neq\mathcal{L}_{\mathcal{T}\cup\{i\}}^{-1}(i)}} \mathrm{Card}\big(([|\mathcal{L}_{\mathcal{T}}|+1]\setminus\{u_{\mathcal{L}_{\mathcal{T}\cup\{i\}}^{-1}(i)}\})
\end{aligned}$$

$$(-1)^{-\overline{\mathrm{Ind}}_{\mathcal{T}}^{-1}(k)-\overline{\mathrm{Ind}}_{\mathcal{T}}^{-1}(k')+\sum_{i_1\in[|\mathcal{L}_{\mathcal{T}}|]}\mathrm{Card}([|\mathcal{L}_{\mathcal{T}}|]\setminus\{u_1,\ldots,u_{i_1}\},u_{i_1})+\sum_{i'_1\in[|\mathcal{L}_{\mathcal{T}}|]}\mathrm{Card}([|\mathcal{L}_{\mathcal{T}}|]\setminus\{u'_1,\ldots,u'_{i'_1}\},u'_{i'_1})}=-1. \qquad (114)$$

$$(-1)^{\sum_{i_1\in[|\mathcal{L}_{\mathcal{T}}|]}\text{Card}([|\mathcal{L}_{\mathcal{T}}|]\setminus\{u_1,\ldots,u_{i_1}\},u_{i_1})+\sum_{i'_1\in[|\mathcal{L}_{\mathcal{T}}|]}\text{Card}([|\mathcal{L}_{\mathcal{T}}|]\setminus\{u'_1,\ldots,u'_{i'_1}\},u'_{i'_1})}$$
$$=(-1)^{\text{Card}([|\mathcal{L}_{\mathcal{T}}|],u_{\mathcal{L}_{\mathcal{T}}^{-1}(i)})+\text{Card}([|\mathcal{L}_{\mathcal{T}}|],u'_{\mathcal{L}_{\mathcal{T}}^{-1}(i)})}. \tag{118}$$

$$(-1)^{\sum_{i_1\in[|\mathcal{L}_{\mathcal{T}}|]}\text{Card}([|\mathcal{L}_{\mathcal{T}}|]\setminus\{u_1,\ldots,u_{i_1}\},u_{i_1})+\sum_{i'_1\in[|\mathcal{L}_{\mathcal{T}}|]}\text{Card}([|\mathcal{L}_{\mathcal{T}}|]\setminus\{u'_1,\ldots,u'_{i'_1}\},u'_{i'_1})}$$
$$=(-1)^{\overline{\text{Ind}}_{\mathcal{T}}^{-1}(k')-2+\overline{\text{Ind}}_{\mathcal{T}}^{-1}(k)-1} \tag{123a}$$
$$=(-1)^{\overline{\text{Ind}}_{\mathcal{T}}^{-1}(k')+\overline{\text{Ind}}_{\mathcal{T}}^{-1}(k)+1}. \tag{123b}$$

$$(-1)^{-\overline{\text{Ind}}_{\mathcal{T}}^{-1}(k)-\overline{\text{Ind}}_{\mathcal{T}}^{-1}(k')+\sum_{i_1\in[|\mathcal{L}_{\mathcal{T}}|]}\text{Card}([|\mathcal{L}_{\mathcal{T}}|]\setminus\{u_1,\ldots,u_{i_1}\},u_{i_1})+\sum_{i'_1\in[|\mathcal{L}_{\mathcal{T}}|]}\text{Card}([|\mathcal{L}_{\mathcal{T}}|]\setminus\{u'_1,\ldots,u'_{i'_1}\},u'_{i'_1})}$$
$$=(-1)^{-\overline{\text{Ind}}_{\mathcal{T}}^{-1}(k)-\overline{\text{Ind}}_{\mathcal{T}}^{-1}(k')+\overline{\text{Ind}}_{\mathcal{T}}^{-1}(k')+\overline{\text{Ind}}_{\mathcal{T}}^{-1}(k)+1} \tag{124a}$$
$$=-1, \tag{124b}$$

$$(-1)^{1+\mathcal{L}_{\mathcal{T}\cup\{i\}}^{-1}(i)-\overline{\text{Ind}}_{\mathcal{T}}^{-1}(\widetilde{k})+\sum_{i_1\in[|\mathcal{L}_{\mathcal{T}}|+1]}\text{Card}([|\mathcal{L}_{\mathcal{T}}|+1]\setminus\{u_1,\ldots,u_{i_1}\},u_{i_1})+\sum_{\widetilde{i}_1\in[|\mathcal{L}_{\mathcal{T}}|]}\text{Card}([|\mathcal{L}_{\mathcal{T}}|]\setminus\{\widetilde{u}_1,\ldots,\widetilde{u}_{\widetilde{i}_1}\},\widetilde{u}_{\widetilde{i}_1})}=-1. \tag{125}$$

$$(-1)^{1+\mathcal{L}_{\mathcal{T}\cup\{i\}}^{-1}(i)-\overline{\text{Ind}}_{\mathcal{T}}^{-1}(\widetilde{k})+\sum_{i_1\in[|\mathcal{L}_{\mathcal{T}}|+1]}\text{Card}([|\mathcal{L}_{\mathcal{T}}|+1]\setminus\{u_1,\ldots,u_{i_1}\},u_{i_1})+\sum_{\widetilde{i}_1\in[|\mathcal{L}_{\mathcal{T}}|]}\text{Card}([|\mathcal{L}_{\mathcal{T}}|]\setminus\{\widetilde{u}_1,\ldots,\widetilde{u}_{\widetilde{i}_1}\},\widetilde{u}_{\widetilde{i}_1})}$$
$$=(-1)^{-\overline{\text{Ind}}_{\mathcal{T}}^{-1}(\widetilde{k})+\text{Card}([|\mathcal{L}_{\mathcal{T}}|+1],u_{\mathcal{L}_{\mathcal{T}\cup\{i\}}^{-1}(i)})} \tag{128a}$$
$$=-1, \tag{128b}$$

$$\setminus\{u_1,\ldots,u_{\mathcal{L}_{\mathcal{T}\cup\{i\}}^{-1}(i)-1},u_{\mathcal{L}_{\mathcal{T}\cup\{i\}}^{-1}(i)+1},\ldots,u_{i_1}\},u_{i_1}\big)$$
$$+\mathcal{L}_{\mathcal{T}\cup\{i\}}^{-1}(i)-1-|\{i_2\in[\mathcal{L}_{\mathcal{T}\cup\{i\}}^{-1}(i)-1]:u_{i_2}<u_{\mathcal{L}_{\mathcal{T}\cup\{i\}}^{-1}(i)}\}|$$
$$+\text{Card}([|\mathcal{L}_{\mathcal{T}}|+1],u_{\mathcal{L}_{\mathcal{T}\cup\{i\}}^{-1}(i)})$$
$$-|\{i_2\in[\mathcal{L}_{\mathcal{T}\cup\{i\}}^{-1}(i)-1]:u_{i_2}<u_{\mathcal{L}_{\mathcal{T}\cup\{i\}}^{-1}(i)}\}| \tag{126b}$$

From the construction of $\widetilde{\mathbf{u}}$ in (105a), we have

$$\sum_{\substack{i_1\in[|\mathcal{L}_{\mathcal{T}}|+1]:\\ i_1\neq\mathcal{L}_{\mathcal{T}\cup\{i\}}^{-1}(i)}}\text{Card}\big(([|\mathcal{L}_{\mathcal{T}}|+1]\setminus\{u_{\mathcal{L}_{\mathcal{T}\cup\{i\}}^{-1}(i)}\})$$
$$\setminus\{u_1,\ldots,u_{\mathcal{L}_{\mathcal{T}\cup\{i\}}^{-1}(i)-1},u_{\mathcal{L}_{\mathcal{T}\cup\{i\}}^{-1}(i)+1},\ldots,u_{i_1}\},u_{i_1}\big)$$
$$=\sum_{\widetilde{i}_1\in[|\mathcal{L}_{\mathcal{T}}|]}\text{Card}([|\mathcal{L}_{\mathcal{T}}|]\setminus\{\widetilde{u}_1,\ldots,\widetilde{u}_{\widetilde{i}_1}\},\widetilde{u}_{\widetilde{i}_1}). \tag{127}$$

From (126b) and (127), and the fact that $(-1)^{2a}=(-1)^0$ for any integer a, we have, (128), shown at the top of the page.

where (128b) comes from that $\widetilde{k}:=\overline{\text{Ind}}_{\mathcal{T}}(u_{\mathcal{L}_{\mathcal{T}\cup\{i\}}^{-1}(i)})$, and thus $u_{\mathcal{L}_{\mathcal{T}\cup\{i\}}^{-1}(i)}=\overline{\text{Ind}}_{\mathcal{T}}^{-1}(\widetilde{k})$

REFERENCES

[1] K. Wan, H. Sun, M. Ji, D. Tuninetti, and G. Caire, "Cache-aided scalar linear function retrieval," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2020, pp. 1717–1722.

[2] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2856–2867, May 2014.

[3] K. Wan, D. Tuninetti, and P. Piantanida, "On the optimality of uncoded cache placement," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Sep. 2016, pp. 161–165.

[4] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "The exact rate-memory tradeoff for caching with uncoded prefetching," *IEEE Trans. Inf. Theory*, vol. 64, no. 2, pp. 1281–1296, Feb. 2018.

[5] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "Characterizing the rate-memory tradeoff in cache networks within a factor of 2," *IEEE Trans. Inf. Theory*, vol. 65, no. 1, pp. 647–663, Jan. 2019.

[6] M. Ji, G. Caire, and A. F. Molisch, "Fundamental limits of caching in wireless D2D networks," *IEEE Trans. Inf. Theory*, vol. 62, no. 2, pp. 849–869, Feb. 2016.

[7] F. Engelmann and P. Elia, "A content-delivery protocol, exploiting the privacy benefits of coded caching," in *Proc. 15th Int. Symp. Modeling Optim. Mobile, Ad Hoc, Wireless Netw. (WiOpt)*, May 2017, pp. 1–6.

[8] K. Wan and G. Caire, "On coded caching with private demands," *IEEE Trans. Inf. Theory*, vol. 67, no. 1, pp. 358–372, Jan. 2021.

[9] K. Wan, H. Sun, M. Ji, D. Tuninetti, and G. Caire, "Device-to-device private caching with trusted server," 2019, *arXiv:1909.12748*. [Online]. Available: http://arxiv.org/abs/1909.12748

[10] S. Li, M. A. Maddah-Ali, Q. Yu, and A. S. Avestimehr, "A fundamental tradeoff between computation and communication in distributed computing," *IEEE Trans. Inf. Theory*, vol. 64, no. 1, pp. 109–128, Jan. 2018.

[11] M. Adel Attia and R. Tandon, "Near optimal coded data shuffling for distributed learning," *IEEE Trans. Inf. Theory*, vol. 65, no. 11, pp. 7325–7349, Nov. 2019.

[12] A. Elmahdy and S. Mohajer, "On the fundamental limits of coded data shuffling for distributed machine learning," *IEEE Trans. Inf. Theory*, vol. 66, no. 5, pp. 3098–3131, May 2020.

[13] K. Wan, D. Tuninetti, M. Ji, G. Caire, and P. Piantanida, "Fundamental limits of decentralized data shuffling," *IEEE Trans. Inf. Theory*, vol. 66, no. 6, pp. 3616–3637, Jun. 2020.
[14] M. Ji, A. Tulino, J. Llorca, and G. Caire, "Caching-aided coded multicasting with multiple random requests," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Apr. 2015, pp. 1–5.
[15] A. Sengupta and R. Tandon, "Improved approximation of storage-rate tradeoff for caching with multiple demands," *IEEE Trans. Commun.*, vol. 65, no. 5, pp. 1940–1955, May 2017.
[16] J. So, B. Guler, and A. S. Avestimehr, "CodedPrivateML: A fast and privacy-preserving framework for distributed machine learning," *IEEE J. Sel. Areas Inf. Theory*, early access, Jan. 21, 2021, doi: 10.1109/JSAIT.2021.3053220.
[17] M. Aliasgari, O. Simeone, and J. Kliewer, "Private and secure distributed matrix multiplication with flexible communication load," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 2722–2734, 2020.
[18] K. Shanmugam, M. Ji, A. M. Tulino, J. Llorca, and A. G. Dimakis, "Finite-length analysis of caching-aided coded multicasting," *IEEE Trans. Inf. Theory*, vol. 62, no. 10, pp. 5524–5537, Oct. 2016.
[19] S. Jin, Y. Cui, H. Liu, and G. Caire, "A new order-optimal decentralized coded caching scheme with good performance in the finite file size regime," *IEEE Trans. Commun.*, vol. 67, no. 8, pp. 5297–5310, Aug. 2019.
[20] M. Cheng, J. Jiang, Q. Wang, and Y. Yao, "A generalized grouping scheme in coded caching," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3422–3430, May 2019.
[21] Q. Yan, M. Cheng, X. Tang, and Q. Chen, "On the placement delivery array design for centralized coded caching scheme," *IEEE Trans. Inf. Theory*, vol. 63, no. 9, pp. 5821–5833, Sep. 2017.
[22] X. Zhong, M. Cheng, and J. Jiang, "Placement delivery array based on concatenating construction," *IEEE Commun. Lett.*, vol. 24, no. 6, pp. 1216–1220, Jun. 2020.
[23] M. Cheng, J. Li, X. Tang, and R. Wei, "Linear coded caching scheme for centralized networks," *IEEE Trans. Inf. Theory*, vol. 67, no. 3, pp. 1732–1742, Mar. 2021.
[24] K. Shanmugam, A. M. Tulino, and A. G. Dimakis, "Coded caching with linear subpacketization is possible using Ruzsa-Szeméredi graphs," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 1237–1241.
[25] L. Tang and A. Ramamoorthy, "Coded caching schemes with reduced subpacketization from linear block codes," *IEEE Trans. Inf. Theory*, vol. 64, no. 4, pp. 3099–3120, Apr. 2018.
[26] S. Agrawal, K. V. Sushena Sree, and P. Krishnan, "Coded caching based on combinatorial designs," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2019, pp. 1227–1231.
[27] H. H. S. Chittoor, M. Bhavana, and P. Krishnan, "Coded caching via projective geometry: A new low subpacketization scheme," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2019, pp. 682–686.
[28] H. Sun and S. A. Jafar, "The capacity of private computation," *IEEE Trans. Inf. Theory*, vol. 65, no. 6, pp. 3880–3897, Jun. 2019.
[29] Y. Yakimenka, H.-Y. Lin, and E. Rosnes, "On the capacity of private monomial computation," 2020, *arXiv:2001.06320*. [Online]. Available: http://arxiv.org/abs/2001.06320
[30] C. Yapar, K. Wan, R. F. Schaefer, and G. Caire, "On the optimality of D2D coded caching with uncoded cache placement and one-shot delivery," *IEEE Trans. Commun.*, vol. 67, no. 12, pp. 8179–8192, Dec. 2019.
[31] Q. Yan and D. Tuninetti, "Fundamental limits of caching for demand privacy against colluding users," *IEEE J. Sel. Areas Inf. Theory*, early access, Jan. 21, 2021, doi: 10.1109/JSAIT.2021.3053372.

Kai Wan (Member, IEEE) received the B.E. degree in optoelectronics from the Huazhong University of Science and Technology, China, in 2012, and the M.Sc. and Ph.D. degrees in communications from Université Paris Saclay, France, in 2014 and 2018, respectively. He is currently a Post-Doctoral Researcher with the Communications and Information Theory Chair (CommIT), Technische Universität Berlin, Berlin, Germany. His research interests include information theory, coding techniques, and their applications on coded caching, index coding, distributed storage, distributed computing, wireless communications, privacy, and security.

Hua Sun (Member, IEEE) received the B.E. degree in communications engineering from the Beijing University of Posts and Telecommunications, China, in 2011, and the M.S. degree in electrical and computer engineering and the Ph.D. degree in electrical engineering from the University of California at Irvine, USA, in 2013 and 2017, respectively.

He is currently an Assistant Professor with the Department of Electrical Engineering, University of North Texas, USA. His research interests include information theory and its applications to communications, privacy, security, and storage. He was a recipient of the NSF CAREER Award in 2021. His coauthored papers received the IEEE Jack Keil Wolf ISIT Student Paper Award in 2016 and the IEEE GLOBECOM Best Paper Award in 2016.

Mingyue Ji (Member, IEEE) received the B.E. degree in communication engineering from the Beijing University of Posts and Telecommunications, China, in 2006, the M.Sc. degree in electrical engineering from the Royal Institute of Technology, Sweden, in 2008, the M.Sc. degree in electrical engineering from the University of California at Santa Cruz, Santa Cruz, in 2010, and the Ph.D. degree from the Ming Hsieh Department of Electrical Engineering, University of Southern California, in 2015. He was subsequently a Staff II System Design Scientist with Broadcom Corporation (Broadcom Ltd.) from 2015 to 2016. He is currently an Assistant Professor with the Electrical and Computer Engineering Department, The University of Utah. His research interests include information theory, coding theory, concentration of measure and statistics with the applications of caching networks, wireless communications, distributed storage and computing systems, federated learning, and (statistical) signal processing. He received the IEEE Communications Society Leonard G. Abraham Prize 2019, the Best Paper Award in IEEE ICC 2015 Conference, the Best Student Paper Award in IEEE European Wireless 2010 Conference, and the USC Annenberg Fellowship from 2010 to 2014. He is also an Associate Editor of IEEE TRANSACTIONS ON COMMUNICATIONS.

Daniela Tuninetti (Fellow, IEEE) received the Ph.D. degree in electrical engineering from ENST/Télécom ParisTech, Paris, France, in 2002, with work done at the Eurecom Institute, Sophia Antipolis, France. She is currently a Professor with the Department of Electrical and Computer Engineering, University of Illinois at Chicago (UIC), where she joined in 2005. She was a Post-Doctoral Research Associate with the School of Communication and Computer Science, Swiss Federal Institute of Technology in Lausanne (EPFL), Lausanne, Switzerland, from 2002 to 2004. Her research interests include the ultimate performance limits of wireless interference networks (with special emphasis on cognition and user cooperation), coexistence between radar and communication systems, multi-relay networks, content-type coding, cache-aided systems, and distributed private coded computing. She was a recipient of the Best Paper Award at the European Wireless Conference in 2002, the NSF CAREER Award in 2007, and named as the University of Illinois Scholar in 2015. She was the Editor-in-Chief of the IEEE Information Theory Society Newsletter from 2006 to 2008, an Editor of IEEE COMMUNICATIONS LETTERS from 2006 to 2009, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS from 2011 to 2014, and IEEE TRANSACTIONS ON INFORMATION THEORY from 2014 to 2017. She is also an Editor of IEEE TRANSACTIONS ON COMMUNICATIONS. She is also a Distinguished Lecturer of the Information Theory Society.

Giuseppe Caire (Fellow, IEEE) was born in Torino in 1965. He received the B.Sc. degree in electrical engineering from the Politecnico di Torino in 1990, the M.Sc. degree in electrical engineering from Princeton University in 1992, and the Ph.D. degree from the Politecnico di Torino in 1994.

He was a Post-Doctoral Research Fellow with the European Space Agency (ESTEC), Noordwijk, The Netherlands, from 1994 to 1995, an Assistant Professor of telecommunications with the Politecnico di Torino, an Associate Professor with the University of Parma, Italy, a Professor with the Department of Mobile Communications, Eurecom Institute, Sophia-Antipolis, France, and a Professor of electrical engineering with the Viterbi School of Engineering, University of Southern California, Los Angeles. He is currently an Alexander von Humboldt Professor with the Faculty of Electrical Engineering and Computer Science, Technical University of Berlin, Germany. His main research interests include communications theory, information theory, and channel and source coding, with particular focus on wireless communications. He received the Jack Neubauer Best System Paper Award from the IEEE Vehicular Technology Society in 2003, the IEEE Communications Society and Information Theory Society Joint Paper Award in 2004 and 2011, the Okawa Research Award in 2006, the Alexander von Humboldt Professorship in 2014, the Vodafone Innovation Prize in 2015, the ERC Advanced Grant in 2018, the Leonard G. Abraham Prize for Best IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS Paper in 2019, and the IEEE Communications Society Edwin Howard Armstrong Achievement Award in 2020. He was a recipient of the 2021 Leibinz Prize of the German National Science Foundation (DFG). He has served in the Board of Governors of the IEEE Information Theory Society from 2004 to 2007, and as an Officer from 2008 to 2013. He was the President of the IEEE Information Theory Society in 2011.