Centralized Caching and Delivery of Correlated Contents Over Gaussian Broadcast Channels

Qianqian Yang[ID], *Member, IEEE*, Parisa Hassanzadeh[ID], Deniz Gündüz[ID], *Senior Member, IEEE*, and Elza Erkip[ID], *Fellow, IEEE*

Abstract—Content delivery in a multi-user cache-aided broadcast network is studied, where a server holding a database of correlated contents communicates with the users over a Gaussian broadcast channel (BC). The minimum transmission power required to satisfy all possible demand combinations is studied, when the users are equipped with caches of equal size. Two centralized caching schemes are proposed, both of which not only utilize the user's local caches, but also exploit the correlation among the contents in the database. The first scheme implements uncoded cache placement and delivers coded contents to users using superposition coding. The second scheme, which is proposed for small cache sizes, places coded contents in users' caches and jointly encodes the cached contents of users and the messages targeted at them. The performance of the proposed schemes, which provide upper bounds on the required transmit power for a given cache capacity, is characterized. The scheme based on coded placement improves upon the first one for small cache sizes, and under certain conditions meets the uncoded placement lower bound. A lower bound on the required transmit power is also presented assuming uncoded cache placement. Our results indicate that exploiting the correlations among the contents in a cache-aided Gaussian BC can provide significant energy savings.

Index Terms—Coded caching, noisy broadcast channel, content correlation.

I. INTRODUCTION

THANKS to the decreasing cost and increasing capacity of storage available at mobile devices, *proactive caching* has received significant attention in recent years as a low-cost and effective solution to keep up with the exponentially growing mobile data traffic [2]–[4]. Proactively storing popular contents in cache memories distributed across the network during off-peak traffic periods can greatly reduce both the network congestion and the latency during peak traffic hours. *Coded*
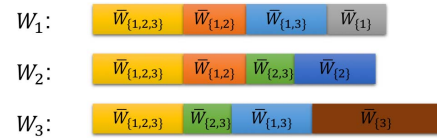
Fig. 1.   An example with $N = 3$ correlated files. Each file consists of 4 different subfiles with different *commonness* levels.

caching [3] exploits the broadcast nature of wireless delivery and the contents proactively cached in users' local memories to create multicasting opportunities, even when the users request distinct files, further boosting the benefits of caching. The significant gains of coded caching over traditional uncoded caching schemes have inspired numerous studies, among which [5]–[15] are most related to this paper.

Most of the literature on coded caching considers independent files in the library. However, in many practical settings, files in a cache library can be highly correlated. For example, if we treat chunks of a video file as distinct files to be cached and delivered, these video chunks are typically correlated. Similarly when delivering software updates, each user may request a different version, or updates for a different subset of software packages, which may lead to correlations among requests. In the file correlation model, used in this paper and introduced in [6], we assume that any subset of the files in the library exclusively share a common part. We present an example of the considered correlation model for three files (under the usual information theoretic assumptions of sufficiently large blocklengths, or equivalently, sufficiently large file sizes in our model) in Fig. 1, where the common parts of different subsets of files are shown with different colors. This model is fairly general to capture message correlations on the symbol level modeled by arbitrary joint distributions, as more commonly considered in multi-terminal source coding problems [16], when it is used in conjunction with the Gray-Wyner network [17], which, as described in [18], encodes the correlated files into messages with the correlation structure considered in this paper. The same message correlation structure has also been considered in the study of multi-access channels with correlated sources in [19], where each channel input terminal observes a predetermined nonempty subset of independent messages.

Delivering correlated contents over an error-free shared link with receiver caches is considered in [5]–[8]. In [5], correlations among an arbitrary number of files is exploited by identifying the most representative files, which are then used as references for compressing the remaining files with respect

to the representatives. Correlation among two files is fully exploited in [8], in which the files are initially compressed using Gray-Wyner source coding, and an optimal caching scheme is derived for the two-receiver network. This scheme is generalized to more files in [7], which is optimal for large cache sizes. Arbitrary numbers of users and files are considered in [6], with the file correlation model illustrated in Fig. 1.

The works in [11]–[15] consider a more realistic noisy broadcast channel (BC) model from the server to the user. In [14], the authors consider a degraded BC and a total memory budget, and optimize the cache assignment to the users depending on their channel capacities. A different perspective is taken in [12], which highlights the benefits of caching and coded delivery in terms of the energy-efficiency in a Gaussian BC. However, neither of these papers consider correlation among files.

Following upon [12], in this paper we consider a degraded Gaussian BC model, but rather than independent files, we assume that the files in the library can be arbitrarily correlated as modeled in [6], and illustrated in Fig. 1. In this model, we have a total of $2^N - 1$ subfiles (which can be of size zero), each shared exclusively by a distinct subset of users. We evaluate the performance of this system in terms of the minimum transmission power required to satisfy any demand combination. We propose two different caching and delivery schemes for this system. The first scheme employs uncoded cache placement, superposition coding and power allocation for the delivery of coded contents. For small cache sizes, a second scheme based on coded placement and joint encoding scheme is also considered as coded placement is known to better exploit limited cache capacities [20] and in asymmetric scenarios [21]. The proposed scheme further exploits the degraded nature of the BC channel by jointly encoding the parts of the contents requested by the stronger users that are available in the weak users' caches together with the messages targeted at them. As a result stronger users can receive parts of their requests that are available in the caches of weak users at no additional energy cost, and without causing interference at the weak users. We also derive a lower bound on the transmission power assuming uncoded cache placement. We show that the required transmission power of the scheme that employs coded placement meets the derived lower bound that assumes uncoded placement under certain conditions. Through simulations, we show that the proposed correlation-aware joint caching and channel coding schemes reduce the transmission power significantly compared to correlation-ignorant counterparts.

The paper is organized as follows. Section II presents the system model and the problem formulation. A lower bound is presented in Section V. Two centralized caching and delivery schemes are proposed in Sections III and IV based, respectively, on separate and joint cache-channel coding. Numerical results comparing the proposed upper and lower bounds are provided in Section VI, and the paper is concluded in Section VII.

Notations: The set of integers $\{i, \ldots, j\}$, where $i \leq j$, is denoted by $[i : j]$, and for $q \in \mathbb{R}^+$, the set $[1 : \lceil q \rceil]$ is

denoted shortly by $[q]$. For sets $\mathcal{A}$ and $\mathcal{B}$, we define $\mathcal{A} \backslash \mathcal{B} \triangleq \{x : x \in \mathcal{A}, x \notin \mathcal{B}\}$, and $|\mathcal{A}|$ denotes the cardinality of $\mathcal{A}$. $\binom{j}{i}$ represents the binomial coefficient if $j \geq i$; otherwise, $\binom{j}{i} = 0$. For event E, $\mathbb{1}\{E\} = 1$ if E is true; and $\mathbb{1}\{E\} = 0$, otherwise.

II. SYSTEM MODEL

Consider a server that holds a database of N correlated files, denoted by $\mathbf{W} = (W_1, \ldots, W_N)$, each composed of a group of independent subfiles. File W_i, $i \in [N]$, consists of 2^{N-1} independent subfiles, i.e.,

$$W_i = \{\overline{W}_\mathcal{S} : \mathcal{S} \subseteq [N], \ i \in \mathcal{S}\},$$

where $\overline{W}_\mathcal{S}$ denotes the subfile shared exclusively by the files $\{W_i : i \in \mathcal{S}\}$. For $\mathcal{S} \subseteq [N]$, $|\mathcal{S}| = \ell$, we say that subfile $\overline{W}_\mathcal{S}$ has a *commonness level of* ℓ. The subfiles are arranged into N *sublibraries*, $L_1, \ldots, L_N$, such that L_ℓ contains all the subfiles with commonness level of ℓ, i.e.,

$$L_\ell = \{\overline{W}_\mathcal{S} : \mathcal{S} \subseteq [N], \ |\mathcal{S}| = \ell\}.$$

We assume that all the subfiles with the same commonness level, i.e., in the same sublibrary, have the same length, and let subfile $\overline{W}_\mathcal{S} \in L_\ell$ be distributed uniformly over the set $[2^{nR_\ell}]$, where R_ℓ is referred to as the rate of subfile $\overline{W}_\mathcal{S}$, and n denotes the transmission blocklength, corresponding to n uses of the BC. Let $\mathbf{R} \triangleq (R_1, \ldots, R_N)$. Therefore, all the files are of the same rate of R bits per channel use, given by

$$R = \sum_{\ell=1}^{N} \binom{N-1}{\ell-1} R_\ell.$$

Each user is equipped with a cache of size nM bits, where M is called the *normalized cache capacity*, and identifies how the allocated cache capacity should be scaled with the number of channel uses. Communication takes place in two phases. During the first phase, referred to as the *placement phase*, the user caches are filled by the server without the knowledge of user demands. This phase happens during a period of low traffic, and we assume during that phase the channel is noiseless and there are no rate limitations. We consider *centralized* caching; that is, the server has the knowledge of the active users in advance, allowing the cache placement to be conducted in a coordinated fashion. At the beginning of the second phase, referred to as the *delivery phase*, user $k \in [K]$ requests file W_{d_k} from the library, with d_k uniformly distributed over $[N]$. Let $\mathbf{d} \triangleq (d_1, \ldots, d_K)$ denote the demand vector. All the requests are satisfied through a Gaussian BC, characterized by a time-invariant channel vector $\mathbf{h} = (h_1, \ldots, h_K)$ and additive white Gaussian noise, where h_k denotes the real channel gain between the server and user k. The channel gains are fixed, and are known to all the parties. Without loss of generality, we assume $h_1^2 \leq h_2^2 \leq \cdots \leq h_K^2$, such that the users are ordered from the weakest to the strongest. The ith channel output at user k is given by

$$Y_{k,i} = h_k X_i + \sigma_{k,i},$$

where X_i and $\sigma_{k,i} \sim \mathcal{N}(0, 1)$ denote the channel input and the noise term at user k in the ith channel use, respectively,

which is independent and identically distributed across time and users.

For a total transmit power of P, an $(n, \mathbf{R}, M, P)$ code for this system consists of:

- K **caching functions** f_k, $k \in [K]$,

$$f_k : [2^{nR}]^N \times \mathbb{R}^K \to [2^{nMR}],$$

  such that user k's cache content is given by $Z_k = f_k(\mathbf{W}, \mathbf{h})$. Let $\mathbf{Z} \triangleq (Z_1, \ldots, Z_K)$.

- A **delivery function** g,

$$g : [2^{nR}]^N \times [2^{nMR}] \times \mathbb{R}^K \times [N]^K \to \mathbb{R}^n,$$

  which, for given cache contents $\mathbf{Z}$, channel gains $\mathbf{h}$, and demand vector $\mathbf{d}$, generates the channel input signal, $X^n(\mathbf{W}, \mathbf{Z}, \mathbf{d}) = g(\mathbf{W}, \mathbf{Z}, \mathbf{h}, \mathbf{d})$, transmitted by the server over the Gaussian BC in n channel uses, with $X_i(\mathbf{W}, \mathbf{Z}, \mathbf{d})$ denoting the ith channel input, $i = 1, \ldots, n$. The channel input vector is generated such that its average power over n channel uses is not more than P for any demand vector realization, i.e.,

$$P(\mathbf{W}, \mathbf{Z}, \mathbf{d}) \triangleq \frac{1}{n} \sum_{i=1}^{n} X_i^2(\mathbf{W}, \mathbf{Z}, \mathbf{d}) \leq P, \quad \forall \mathbf{d} \in [N]^K.$$

- K **decoding functions** ϕ_k, $k \in [K]$,

$$\phi_k : \mathbb{R}^n \times [2^{nMR}] \times \mathbb{R}^K \times [N]^K \to [2^{nR}],$$

  where $\widehat{W}_{d_k} = \phi_k(Y^n(\mathbf{W}, \mathbf{Z}, \mathbf{d}), Z_k, \mathbf{h}, \mathbf{d})$, is the reconstruction of W_{d_k} requested by user k, and $Y^n(\mathbf{W}, \mathbf{Z}, \mathbf{d})$ is the channel output at user k for input signal $X^n(\mathbf{W}, \mathbf{Z}, \mathbf{d})$.

Definition 1: A memory-power pair (M, P) is achievable for the system described above, if there exists a sequence of $(n, \mathbf{R}, M, P)$ codes such that

$$\lim_{n \to \infty} \mathbb{P} \left\{ \bigcup_{\mathbf{d} \in [N]^K} \bigcup_{k=1}^{K} \left\{ \widehat{W}_{d_k} \neq W_{d_k} \right\} \right\} = 0.$$

For a system with N files and K users, with given channel gains $\mathbf{h}$, our goal is to characterize the minimum achievable power P as a function of the user cache capacity M, i.e.,

$$P^*(M) \triangleq \inf\{P : (M, P) \text{ is achievable}\}.$$

Remark 1: In principle different codebooks satisfying different average power constraints can be used for different demand vectors. With the definition above, our goal is to characterize the power constraint that is required to satisfy any demand combination.

We conclude this section with the following proposition, which will be frequently referred to in the remainder of the paper.

Proposition 1 [22], [23]: In a K-user degraded Gaussian BC with $h_1^2 \leq h_2^2 \leq \cdots \leq h_K^2$, distinct messages at rates $\rho_1, \ldots, \rho_K$, can be reliably transmitted to users $1, \ldots, K$, respectively, if and only if

$$\rho_k \leq C \left(\frac{h_k^2 P_k}{1 + h_k^2 \sum\limits_{j=k+1}^{K} P_j} \right), \quad k = 1, \ldots, K, \tag{1}$$

where $C(x) \triangleq \frac{1}{2} \log_2(1+x)$. *This is achieved by superposition coding with Gaussian codewords of power P_i, $i = 1, \ldots, K$, to transmit to user i. As a consequence, the minimum total transmit power for reliable communication is given by*

$$\sum_{k=1}^{K} P_k \geq \sum_{k=1}^{K} \left(\frac{2^{2\rho_k} - 1}{h_k^2} \right) \prod_{j=1}^{k-1} 2^{2\rho_j}. \tag{2}$$

III. CACHE-AIDED SUPERPOSITION CODING

We first propose a centralized caching and delivery scheme, which employs superposition coding to deliver coded messages over the Gaussian BC [22], [23], where the coded messages are generated taking into account the correlation among the requested files as well as the channel gains. As in [6]–[8], the scheme operates by treating the sublibraries independently during the placement and delivery phases to determine the cache content and messages targeted at each user, which are then jointly delivered over the BC. For clarity, the scheme is first explained on a simple example.

A. Motivating Example

Example 1: Consider $K = 3$ users with channel gains $h_1^2 \leq h_2^2 \leq h_3^2$, and a database of $N = 3$ files as in Fig. 1 with sublibraries:

- $L_1 = \{\overline{W}_{\{1\}}, \overline{W}_{\{2\}}, \overline{W}_{\{3\}}\}$, each with rate R_1.
- $L_2 = \{\overline{W}_{\{1,2\}}, \overline{W}_{\{2,3\}}, \overline{W}_{\{1,3\}}\}$, each with rate R_2.
- $L_3 = \{\overline{W}_{\{1,2,3\}}\}$, with rate R_3.

Assume that each user has a normalized cache capacity of $M = R_1 + R_2 + \frac{1}{3} R_3$.

○ **Placement Phase:** Placement is carried out independently across sublibraries. Assume that each user divides its cache into three portions with normalized capacities R_1, R_2, and $\frac{1}{3} R_3$, allocated for files from sublibraries L_1, L_2 and L_3, respectively. We remark that this cache capacity allocation is not optimal, and the proposed scheme further optimizes the allocation as described in Sec. III-B. We use the prefetching policy proposed in [10], which divides the subfiles in sublibrary L_ℓ into three non-overlapping parts, each of size $\frac{1}{3} nR_\ell$ bits. Then, user k caches

$$Z_k = \left\{ \overline{W}_{\{1\},\{k\}}, \overline{W}_{\{2\},\{k\}}, \overline{W}_{\{3\},\{k\}}, \overline{W}_{\{1,2\},\{k\}}, \right.$$
$$\left. \overline{W}_{\{2,3\},\{k\}}, \overline{W}_{\{1,3\},\{k\}}, \overline{W}_{\{1,2,3\},\{k\}} \right\},$$

where $\overline{W}_{\mathcal{S},\{k\}}$ denotes the kth part of subfile $\overline{W}_{\mathcal{S}}$ cached at user $k \in [3]$.

○ **Delivery Phase:** Once the demand vector is revealed, the server computes the messages intended for each user, independently for each sublibrary, and delivers them over the BC via superposition coding with Gaussian codewords. The total transmit power is given in Proposition 1, which depends on the rate of messages intended for each user. Consider the demand vector $\mathbf{d} = (1, 2, 3)$. User 1, the weakest user, needs subfiles $\{\overline{W}_{\{1\}}, \overline{W}_{\{1,2\}}, \overline{W}_{\{1,3\}}, \overline{W}_{\{1,2,3\}}\}$ to reconstruct W_1. User 2 requires the four subfiles corresponding to file W_2, but since it has a better channel than user 1, it can decode the messages targeted at user 1 for free. Similarly, user 3 can

decode the messages indented for both of the weaker users. User messages from each sublibrary are determined as follows.

- Sublibrary L_1: Based on the demand vector, all subfiles in L_1 are required by the users. User 1 needs to receive $\overline{W}_{\{1\},\{2\}}$ and $\overline{W}_{\{1\},\{3\}}$, whose targeted message, denoted by $V_{1,\mathbf{d}}(L_1)$, is generated as follows:

$$V_{1,\mathbf{d}}(L_1) = \{\overline{W}_{\{1\},\{2\}} \oplus \overline{W}_{\{2\},\{1\}}, \overline{W}_{\{1\},\{3\}} \\ \oplus \overline{W}_{\{3\},\{1\}}\}. \quad (3)$$

Since user 2 is able to decode its required part $\overline{W}_{\{2\},\{1\}}$ from message $V_{1,\mathbf{d}}(L_1)$, it only needs $\overline{W}_{\{2\},\{3\}}$, which is recovered through the message

$$V_{2,\mathbf{d}}(L_1) = \{\overline{W}_{\{2\},\{3\}} \oplus \overline{W}_{\{3\},\{2\}}\}. \quad (4)$$

User 3 can decode its missing parts from $V_{1,\mathbf{d}}(L_1)$ and $V_{2,\mathbf{d}}(L_1)$, and therefore, $V_{3,\mathbf{d}}(L_1) = \emptyset$. We note that, while the generation of the coded messages for sublibrary L_1 follows similarly to generic coded caching models with a shared common link, we assign them to users starting from the one with the worst channel gain, as the stronger users automatically decode messages destined for weaker users.

- Sublibrary L_2: Each user requires two subfiles from L_2, which can be considered as two separate demands. We can group these demands into two, with only one demand per user in each group, and deliver the demands within each group separately. One possible grouping of L_2 could be $\mathfrak{S}_1 = (\{1,2\},\{1,2\},\{1,3\})$ and $\mathfrak{S}_2 = (\{1,3\},\{2,3\},\{2,3\})$, where $\mathfrak{S}_1$ corresponds to users 1, 2 and 3 requesting subfiles $\overline{W}_{\{1,2\}}$, $\overline{W}_{\{1,2\}}$ and $\overline{W}_{\{1,3\}}$, respectively. Then $V_{k,\mathbf{d}}(L_2) = \{v_k^1, v_k^2\}$, where v_k^i is user k's message corresponding to group $\mathfrak{S}_i$, $i = 1, 2$. Then, for $\mathfrak{S}_1$ we have

$$v_1^1 = \{\overline{W}_{\{1,2\},\{2\}} \oplus \overline{W}_{\{1,2\},\{1\}}, \overline{W}_{\{1,2\},\{3\}} \\ \oplus \overline{W}_{\{1,3\},\{1\}}\}, \quad (5)$$
$$v_2^1 = \{\overline{W}_{\{1,3\},\{2\}} \oplus \overline{W}_{\{1,2\},\{3\}}\}, \quad (6)$$
$$v_3^1 = \emptyset, \quad (7)$$

and for $\mathfrak{S}_2$

$$v_1^2 = \{\overline{W}_{\{1,3\},\{2\}} \oplus \overline{W}_{\{2,3\},\{1\}}, \overline{W}_{\{1,3\},\{3\}} \\ \oplus \overline{W}_{\{2,3\},\{1\}}\}, \quad (8)$$
$$v_2^2 = \{\overline{W}_{\{2,3\},\{2\}} \oplus \overline{W}_{\{2,3\},\{3\}}\}, \quad (9)$$
$$v_3^2 = \emptyset. \quad (10)$$

- Sublibrary L_3: All users require $\overline{W}_{\{1,2,3\}}$, and therefore

$$V_{1,\mathbf{d}}(L_3) = \{\overline{W}_{\{1,2,3\},\{2\}} \oplus \overline{W}_{\{1,2,3\},\{1\}}, \\ \overline{W}_{\{1,2,3\},\{3\}} \oplus \overline{W}_{\{1,2,3\},\{1\}}\}, \quad (11)$$
$$V_{2,\mathbf{d}}(L_2) = V_{3,\mathbf{d}}(L_2) = \emptyset. \quad (12)$$

The messages in (3), (5), (8) and (11) constitute all the messages targeted for user 1, with total rate $\rho_1 = 2/3(R_1 + 2R_2 + R_3)$. Messages (4), (6) and (9) are targeted for user 2 with total rate $\rho_2 = 1/3(R_1 + 2R_2)$, and finally, user 3 can successfully recover its requested file from the messages intended for users

1 and 2, i.e., $\rho_3 = 0$. Based on Proposition 1, the target rates can be delivered to the users with superposition coding of Gaussian codewords satisfying (1), with a minimum power value given in (2).

B. Proposed Scheme

This section presents the proposed centralized caching and delivery scheme, which generalizes the above example to an arbitrary number of users, and achieves the transmit power value claimed in Theorem 1. Similarly to the schemes in [6]–[8], the proposed scheme treats the sublibraries independently: 1) cache capacity is divided among N sublibraries, 2) for each demand realization, the server identifies the messages that need to be delivered to each user, independently across sublibraries, using a modified version of the scheme proposed in [6], and 3) the server employs superposition coding to reliably communicate coded messages over the Gaussian BC.

1) Placement Phase: Cache contents are identified separately for different sublibraries, each with a different level of commonness. Let $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_N)$ denote the cache allocation vector, where $\pi_\ell \in [0, 1]$ denotes the fraction of the normalized cache capacity M allocated to sublibrary L_ℓ, with $\sum_{\ell=1}^N \pi_\ell = 1$. We will later optimize $\boldsymbol{\pi}$ to minimize the required total power. For a given $\boldsymbol{\pi}$, placement for sublibrary L_ℓ is carried out using the prefetching scheme proposed in [10] as follows. Let

$$t_\ell \triangleq \frac{K\pi_\ell M}{\binom{N}{\ell} R_\ell}, \quad t_\ell \in [0, K], \quad (13)$$

which is not necessarily an integer. We address this by memory-sharing among neighboring integer points, $t_\ell^A \triangleq \lfloor t_\ell \rfloor$ and $t_\ell^B \triangleq \lfloor t_\ell \rfloor + 1$, and divide each subfile $\overline{W}_{\mathcal{S}} \in L_\ell$ into two non-overlapping parts. More specifically, $\overline{W}_{\mathcal{S}} = (\overline{W}_{\mathcal{S}}^A, \overline{W}_{\mathcal{S}}^B)$, where $\overline{W}_{\mathcal{S}}^A$ is at rate $(t_\ell^B - t_\ell) R_\ell$, while $\overline{W}_{\mathcal{S}}^B$ is at rate $(t_\ell - t_\ell^A) R_\ell$. The prefetching policy of [10] is implemented separately for $\{\overline{W}_{\mathcal{S}}^A : \mathcal{S} \in L_\ell\}$ and $\{\overline{W}_{\mathcal{S}}^B : \mathcal{S} \in L_\ell\}$. Each $\overline{W}_{\mathcal{S}}^A$ is split into $\binom{K}{t_\ell^A}$ non-overlapping equal-length parts, each of size $n(t_\ell^B - t_\ell) R_\ell / \binom{K}{t_\ell^A}$ bits. These parts are assigned to sets $\mathcal{A} \subseteq [K]$ of size $|\mathcal{A}| = t_\ell^A$. We denote the part assigned to set $\mathcal{A}$ by $\overline{W}_{\mathcal{S},\mathcal{A}}^A$; therefore,

$$\overline{W}_{\mathcal{S}}^A = \{\overline{W}_{\mathcal{S},\mathcal{A}}^A : \mathcal{A} \subseteq [K], |\mathcal{A}| = t_\ell^A\}.$$

Similarly, each $\overline{W}_{\mathcal{S}}^B$ is split into $\binom{K}{t_\ell^B}$ non-overlapping equal-length parts, which are labeled as

$$\overline{W}_{\mathcal{S}}^B = \{\overline{W}_{\mathcal{S},\mathcal{B}}^B : \mathcal{B} \subseteq [K], |\mathcal{B}| = t_\ell^B\}.$$

User k caches parts $\overline{W}_{\mathcal{S},\mathcal{A}}^A$ if $k \in \mathcal{A}$, and parts $\overline{W}_{\mathcal{S},\mathcal{B}}^B$ if $k \in \mathcal{B}$. With this placement strategy, for each subfile in sublibrary L_ℓ, $\binom{K-1}{t_\ell^A-1}$ distinct parts from $\overline{W}_{\mathcal{S}}^A$, and $\binom{K-1}{t_\ell^B-1}$ distinct parts from $\overline{W}_{\mathcal{S}}^B$, are placed in each user's cache, amounting for a total of $nt_\ell R_\ell / K$ bits, which satisfies the capacity constraint of $n\pi_\ell M$ bits.

2) Delivery Phase: Delivering a file from a library of correlated files can be considered as a multiple-demand problem [6]–[8]. For demand vector $\mathbf{d}$, user k needs $\binom{N-1}{\ell-1}$ subfiles from sublibrary L_ℓ. Since the sublibraries are treated independently, message $V_{k,\mathbf{d}}$, targeted at user k, constitutes the messages computed from all the sublibraries, i.e.,

$$V_{k,\mathbf{d}} = \bigcup_{\ell=1}^{N} V_{k,\mathbf{d}}(L_\ell), \tag{14}$$

where $V_{k,\mathbf{d}}(L_\ell)$ denotes the set of messages from sublibrary L_ℓ targeted at user k. They are determined using Algorithm 1, which is based on [6, Algorithms 1, 2]. The main idea is to treat subfiles $\{\overline{W}_\mathcal{S} : d_k \in \mathcal{S}\}$ that are not cached at user k, as different demands. The algorithm operates by partitioning all the requested subfiles from sublibrary L_ℓ into groups, such that each user requires at most one subfile in each group; resulting in a single-demand problem.

Algorithm 1 Generate Messages $\{V_{1,\mathbf{d}}(L_\ell), \dots, V_{K,\mathbf{d}}(L_\ell)\}$

1: $V_{k,\mathbf{d}}(L_\ell) \leftarrow \emptyset, \forall k \in \{1, \dots, K\}$
2: **for** $r = 1, \dots, \ell$ **do**
3:    $\mathcal{W}_r = \{\overline{W}_\mathcal{S} : |S| = \ell, |\mathcal{S} \cap \mathcal{D}| = r\}$
4:    $\mathfrak{S}_1, \dots, \mathfrak{S}_g \leftarrow$ Group $(\mathcal{W}_r, \mathcal{D}, \ell, r)$
5:    **for** $i \in \{1, \dots, g\}$ **do**
6:      $V_1^A, \dots, V_K^A \leftarrow$ Single-Demand $(A, \mathfrak{S}_i, t_\ell^A)$
7:      $V_1^B, \dots, V_K^B \leftarrow$ Single-Demand $(B, \mathfrak{S}_i, t_\ell^B)$
8:      $V_{k,\mathbf{d}}(L_\ell) \leftarrow V_{k,\mathbf{d}}(L_\ell) \cup \{V_k^A, V_k^B\}, \forall k \in \{1, \dots, K\}$
9:    **end for**
10: **end for**

For sublibrary L_ℓ messages, $V_{1,\mathbf{d}}(L_\ell), \dots, V_{K,\mathbf{d}}(L_\ell)$, are generated as follows:

i) Group the requested subfiles: Let $\mathcal{D} \triangleq \{d_1, \dots, d_K\}$ denote the set of distinct demands in $\mathbf{d}$. Consider the following subset of L_ℓ:

$$\mathcal{W} = \{\overline{W}_\mathcal{S} : \mathcal{S} \subseteq \mathcal{D}, |\mathcal{S}| = \ell\}, \tag{15}$$

where each subfile needs to be delivered to at least ℓ users. Note that each user requests $\binom{|\mathcal{D}|-1}{\ell-1}$ subfiles from $\mathcal{W}$. We group subfiles in $\mathcal{W}$ into $\binom{|\mathcal{D}|-1}{\ell-1}$ demand vectors, such that each demand vector represents a single-demand network with K users, i.e., each demand vector includes only one single demand for each user. The proposed grouping process for $\mathcal{W}$ is specified by function GROUP($\mathcal{W}, \mathcal{D}, \ell, r$) with $r = \ell$, which outputs demand vectors, denoted by $(\mathcal{S}_1, \dots, \mathcal{S}_K)$. To reduce the delivery rate, GROUP($\mathcal{W}, \mathcal{D}, \ell, r$) tries to minimize the number of distinct demands within each single-demand network. The idea is to deliver a subfile $\overline{W}_\mathcal{S}$ to all the users that require this subfile within one single-demand network, i.e., $d_k \in \mathcal{S}$. The GROUP function starts with randomly choosing a $\overline{W}_\mathcal{S} \in \mathcal{W}$ such that $\mathcal{S} \cap \mathcal{D} \subseteq \mathcal{F}$ ($\mathcal{S} \cap \mathcal{D} = \mathcal{S}$ for given $\mathcal{W}$ in (15)), where initially $\mathcal{F} \triangleq \mathcal{D}$. Then it removes $\mathcal{S}$ from $\mathcal{F}$, and $\overline{W}_\mathcal{S}$ from $\mathcal{W}$, and assigns $\mathcal{S}_k = \mathcal{S}$ if $d_k \in \mathcal{S}$. If $|\mathcal{D}|$ is a multiple of ℓ, the GROUP function continues the above procedure until

1: **function** GROUP ($\mathcal{W}, \mathcal{D}, \ell, r$)
2: **Output:** Demand Vectors $\mathfrak{S}_1, \dots, \mathfrak{S}_g$
3:    $\mathcal{F} \leftarrow \mathcal{D}, \overline{\mathcal{F}} \leftarrow \emptyset, \overline{\mathcal{S}} \leftarrow \emptyset, g = 0$
4:    **while** $\mathcal{W} \neq \emptyset$ **do**
5:     **while** $\mathcal{F} \neq \emptyset$ **do**
6:      **if** $|\mathcal{F}| \geq r$ **then**
7:       **if** $\overline{\mathcal{F}} = \emptyset$ **then**
8:        Randomly pick $\overline{W}_\mathcal{S} \in \mathcal{W}$ such that $\mathcal{S} \cap \mathcal{D} \subseteq \mathcal{F}$
9:        $\mathcal{W} \leftarrow \mathcal{W}/\overline{W}_\mathcal{S}, \quad \mathcal{F} \leftarrow \mathcal{F} \setminus \mathcal{S}$
10:        **for** $d_k \in \mathcal{S} \cap \mathcal{D}$ **do**
11:         $\mathcal{S}_k \leftarrow \mathcal{S}$
12:        **end for**
13:       **else**
14:        **for** $d_k \in \overline{\mathcal{F}}$ **do**
15:         $\mathcal{S}_k \leftarrow \overline{\mathcal{S}}$
16:        **end for**
17:        $\mathcal{F} \leftarrow \mathcal{F} \setminus \overline{\mathcal{F}}, \quad \overline{\mathcal{S}} \leftarrow \emptyset, \quad \overline{\mathcal{F}} \leftarrow \emptyset,$
18:       **end if**
19:      **else**
20:       Randomly pick $\overline{W}_\mathcal{S} \in \mathcal{W}$ such that $\mathcal{F} \subset \mathcal{S}$
21:       **for** $d_k \in \mathcal{F}$ **do**
22:        $\mathcal{S}_k \leftarrow \mathcal{S}$
23:       **end for**
24:       $\mathcal{F} \leftarrow \emptyset, \quad \overline{\mathcal{S}} \leftarrow \mathcal{S}, \quad \overline{\mathcal{F}} \leftarrow \mathcal{S} \setminus \mathcal{F}$
25:      **end if**
26:     **end while**
27:     $g = g + 1$
28:     $\mathfrak{S}_g = (\mathcal{S}_1, \dots, \mathcal{S}_K)$
29:     $\mathcal{F} \leftarrow \mathcal{D}$
30:    **end while**
31: **end function**

$\mathcal{F} = \emptyset$ and outputs the corresponding demand vector. This is repeated until $\mathcal{W}$ is empty and all the $\binom{|\mathcal{D}|-1}{\ell-1}$ demand vectors are obtained. In contrast, if $|\mathcal{D}|$ is not a multiple of ℓ, we have $|\mathcal{F}| < \ell$. In this case, the GROUP function randomly chooses a $\overline{W}_\mathcal{S} \in \mathcal{W}$ such that $\mathcal{F} \subset \mathcal{S}$ and $\mathcal{S}_k = \mathcal{S}$ if $d_k \in \mathcal{F}$. Note that any user k for which $d_k \in \mathcal{S} \setminus \mathcal{F}$ also requests $\overline{W}_\mathcal{S}$. Therefore, for the next demand vector, the GROUP function first assigns the demands of these users to be $\mathcal{S}$. It is obvious that there are at most $\lceil |\mathcal{D}|/\ell \rceil + 1$ distinct demands in each demand vector.

Note that, the subfiles in (15) are not the only contents that need to be delivered from sublibrary L_ℓ. Based on the demand vector, any subfile $\overline{W}_\mathcal{S}$ whose index $\mathcal{S}$ includes at least one of the indices in $\mathcal{D}$, i.e., $\mathcal{S} \cap \mathcal{D} \neq \emptyset$, is required for the lossless reconstruction of the corresponding requested file in $\mathcal{D}$. All such subfiles need to be identified, and grouped in a similar fashion. Subfiles in (15) correspond to $|\mathcal{S} \cap \mathcal{D}| = \ell$. For $r = 1, \dots, \ell$, we define the requested subfiles $\mathcal{W}_r$, as

$$\mathcal{W}_r \triangleq \{\overline{W}_\mathcal{S} : |\mathcal{S}| = \ell, |\mathcal{S} \cap \mathcal{D}| = r\}.$$

Then, each $\mathcal{W}_r$ is grouped using the function GROUP in Algorithm 1, which assigns a demand vector $\mathfrak{S}_i = (\mathcal{S}_1, \dots, \mathcal{S}_K)$ to each group, resulting in a

1: **function** SINGLE-DEMAND $(C, \mathfrak{S}, t)$
2: **Input:** $\mathfrak{S} = (\mathcal{S}_1, \ldots, \mathcal{S}_K)$, $C \equiv \{\overline{W}_{\mathcal{S},\mathcal{C}}^C\}$
3: **Output:** Coded messages $V_1, \ldots, V_K$
4:    $\mathcal{K} \leftarrow \{k : \mathcal{S}_k \notin \{\mathcal{S}_1, \ldots, \mathcal{S}_{k-1}\}\}$
5:    **for** $k \in \{1, \ldots, K\}$ **do**
6:      **for** $\mathcal{U} \subseteq [k+1 : K] : |\mathcal{U}| = t, \sum_{j \in \mathcal{K}} \mathbb{1}\{j \in \mathcal{U} \cup \{k\}\} \geq 1$
     do
7:        $V_k \leftarrow V_k \cup \left(\bigoplus_{j \in \mathcal{U} \cup \{k\}} \overline{W}_{\mathcal{S}_j, \mathcal{U} \cup \{k\} \setminus \{j\}}^C \right)$
8:      **end for**
9:    **end for**
10: **end function**

single-demand network with K users, where user k requests subfile $\overline{W}_{\mathcal{S}_k}$.

ii) Deliver the demands corresponding to each group: The groups formed above are treated independently in the delivery phase. More specifically, for a group with corresponding demand vector $\mathfrak{S}$, function SINGLE-DEMAND in Algorithm 1 identifies messages $V_1, \ldots, V_K$ that need to be transmitted so that all the users recover their requested subfiles in $\mathfrak{S}$. These messages are computed using the scheme in [10], and delivered over the degraded BC using the coding scheme in [12]. The channel is taken into account by selecting the *weakest* users with distinct demands as *leaders*, i.e., the demand of a leader is not requested by any of the weaker users, $\{k : \mathcal{S}_k \notin \{\mathcal{S}_1, \ldots, \mathcal{S}_{k-1}\}\}$, and then greedily broadcasting XORed messages that benefit at least one leader through superposition coding. Note that choosing the weakest user, among users requiring the same subfile $\overline{W}_{\mathcal{S}}$, as the leader, allows all the stronger users to decode the subfile through successive cancellation decoding. As mentioned previously, the proposed scheme uses memory-sharing to cache and deliver the subfiles in L_ℓ, for the two parts $\overline{W}_{\mathcal{S}}^A$ and $\overline{W}_{\mathcal{S}}^B$; and therefore, function SINGLE-DEMAND is executed separately for both parts.

Message $V_{k,\mathbf{d}}(L_\ell)$ targeted at user k is the union of all the messages for sublibrary L_ℓ computed for each group identified from the subfile sets $\{\mathcal{W}_1, \ldots, \mathcal{W}_\ell\}$, from which the overall message for user k, $V_{k,\mathbf{d}}$, is obtained by (14). For a given demand vector $\mathbf{d}$, messages $V_{1,\mathbf{d}}, \ldots, V_{K,\mathbf{d}}$ can be reliably transmitted to users $1, \ldots, K$, using a K-level Gaussian superposition codebook [22], [23]. The kth-level codebook consists of $2^{n\rho_k}$ codewords, where ρ_k is the total rate of the messages in $V_{k,\mathbf{d}}$. The total required transmit power is given by (1) in Proposition 1.

C. Achievable Transmit Power

The worst-case transmit power of the scheme described above is presented next.

Theorem 1: For the caching problem described in Section II, the optimal memory-power function, $P^*(M)$,

is upper bounded as

$$P^*(M) \leq \min_{\boldsymbol{\pi} = (\pi_1, \ldots, \pi_N)} P_{UB}(M, \boldsymbol{\pi}),$$

$$\text{s.t.} \sum_{i=1}^{N} \pi_i \leq 1,$$

$$0 \leq \pi_i \leq 1, \quad i = 1, \ldots, N,$$

where

$$P_{UB}(M, \boldsymbol{\pi})$$

$$\triangleq \sum_{k=1}^{K} \left(\frac{2^{2\hat{\rho}_k} - 1}{h_k^2} \right) \prod_{j=1}^{k-1} 2^{2\hat{\rho}_j},$$

$$\hat{\rho}_k \triangleq \sum_{\ell=1}^{N} \sum_{r=\max\{\ell-N+K, 1\}}^{\min\{\ell, K\}} \binom{N-K}{\ell-r}$$

$$\times \binom{\min\{N, K\} - 1}{r - 1} \gamma_{k, \ell, r},$$

$$\gamma_{k, \ell, r} \triangleq \begin{cases} \left(\frac{\binom{K-k}{\lfloor t_\ell \rfloor}}{\binom{K}{\lfloor t_\ell \rfloor}} (\lfloor t_\ell \rfloor + 1 - t) + \frac{\binom{K-k}{\lfloor t_\ell \rfloor + 1}}{\binom{K}{\lfloor t_\ell \rfloor + 1}} (t - \lfloor t_\ell \rfloor) \right) R_\ell, \\ \quad \text{if } k \in [\lceil \frac{\min\{N, K\}}{r} \rceil + 1], \\ 0 \quad \text{otherwise} \end{cases}$$

$$t_\ell \triangleq \frac{K \pi_\ell M}{\binom{N}{\ell} R_\ell}.$$

Proof: This transmit power is achieved by the coding scheme outlined in Algorithm 1. A detailed proof is given in Appendix B. $\square$

IV. CODED PLACEMENT AND JOINT ENCODING

We propose an alternative joint cache-channel coding scheme, with coded placement which is more effective for small cache sizes. The scheme operates by constructing a multi-level superposition code, based on the demand realization, and piggyback part of the messages targeted at each user on the messages intended for weaker users. The piggyback coding is also employed in [14], where all the cache capacity allowance is assigned to the weakest user, and in the delivery phase, part of the content required by each user is piggy-backed onto the message sent to the weakest user. We extend this scheme in two ways: the coded placement is implemented instead of uncoded placement, and the piggyback coding is applied to each layer of superposition code instead of just the first layer. Before presenting the general scheme description, we first provide a brief overview of cache-aided superposition coding, and then use an example to illustrate how part of the messages required by a stronger user can be piggy-backed onto the messages targeted at weaker users.

A. Preliminaries

We extend the piggyback coding in [14] to the case when each user has cached contents. In a cache-aided K-user degraded Gaussian BC with $h_1^2 \leq h_2^2 \leq \cdots \leq h_K^2$, where message V_k^r, with rate ρ_k^r, is locally available at user $k \in [K]$, message V_k^c, with rate ρ_k^c, can be reliably transmitted to user

k, and message $V_k = (V_k^r, V_k^c)$, with rate $\rho_k = \rho_k^r + \rho_k^c$, can be decoded by users $k+1, \ldots, K$, using K-level superposition coding as follows:

- *Codebook construction*: The kth level codebook, denoted by $\mathcal{C}_k$, consists of $\lfloor 2^{n\rho_k^r} \rfloor \times \lfloor 2^{n\rho_k^c} \rfloor$ codewords of block length n, denoted by $x_k^n(v_k^r, v_k^c)$, $v_k^r \in \lfloor 2^{n\rho_k^r} \rfloor$, $v_k^c \in \lfloor 2^{n\rho_k^c} \rfloor$, which are arranged into $\lfloor 2^{n\rho_k^r} \rfloor$ rows and $\lfloor 2^{n\rho_k^c} \rfloor$ columns. The codewords in $\mathcal{C}_k$ are generated independently and identically distributed (i.i.d.) following $x_{k,i} \sim \mathcal{N}(0, P_k)$, $i \in [n]$.

- *Encoding at server*: For messages $V_1, \ldots, V_K$ targeted at the users, the server transmits the superposition of the K codewords $\sum_{k=1}^{K} x_k^n(V_k^r, V_k^c)$ over the Gaussian BC.

- *Decoding at users*: User $k \in [K]$ receives the channel output

$$Y_k^n = h_k \sum_{k=1}^{K} x_k^n(V_k^r, V_k^c) + \sigma_k^n,$$

and based on Proposition 1, it can successfully decode messages $V_1, \ldots, V_{k-1}$, by using successive decoding if

$$\rho_j^r + \rho_j^c \leq C\left(\frac{h_k^2 P_j}{1 + h_k^2 \sum_{j'=k+1}^{K} P_{j'}} \right), \quad \forall j \in [k-1]. \tag{17}$$

Since user k has access to V_k^r, it can extract the subcodebook $\left\{ x_k^n(V_k^r, v_k^c) : v_k^c \in \lfloor 2^{n\rho_k^c} \rfloor \right\}$ from $\mathcal{C}_k$, and losslessly decode V_k^c, if

$$\rho_k^c \leq C\left(\frac{h_k^2 P_k}{1 + h_k^2 \sum_{j=k+1}^{K} P_j} \right).$$

If (17) holds, users $k+1, \ldots K$ can also decode messages $V_1, \ldots, V_{k-1}$ as they have better channel conditions. However, they do not have access to side information V_k^c, so for them to decode V_k successfully, we need

$$\rho_k^r + \rho_k^c \leq C\left(\frac{h_{k+1}^2 P_k}{1 + h_{k+1}^2 \sum_{j=k+1}^{K} P_j} \right).$$

B. Motivating Example

Example 2: Consider the same model as Example 1 in Section III-A, with the rates of the subfiles in three commonness levels given by $R_3 \leq R_2 \leq R_1$, and normalized cache capacity of $M = R_3$.

○ **Placement Phase:** First, we divide each of the subfiles in L_1 and L_2 into two parts:

- Sublibrary L_1: $\overline{W}_{\{i\}} = (\overline{W}_{\{i\}}^C, \overline{W}_{\{i\}}^U)$, $i \in [3]$, where $\overline{W}_{\{i\}}^C$ has rate R_3 while $\overline{W}_{\{i\}}^U$ has rate $R_1 - R_3$.

- Sublibrary L_2: $\overline{W}_{\mathcal{S}} = (\overline{W}_{\mathcal{S}}^C, \overline{W}_{\mathcal{S}}^U)$, $\overline{W}_{\mathcal{S}} \in L_2$, where $\overline{W}_{\mathcal{S}}^C$ has rate R_3 while $\overline{W}_{\mathcal{S}}^U$ has rate $R_2 - R_3$.

Then, users 1, 2, and 3 cache coded contents as follows:

$$Z_1 = \overline{W}_{\{123\}},$$
$$Z_2 = \overline{W}_{\{12\}}^C \oplus \overline{W}_{\{13\}}^C \oplus \overline{W}_{\{23\}}^C,$$
$$Z_3 = \overline{W}_{\{1\}}^C \oplus \overline{W}_{\{2\}}^C \oplus \overline{W}_{\{3\}}^C,$$

such that the weaker users prefetch a linear combination of the subfiles shared among more files.

○ **Delivery Phase:**

- Codebook construction: For the demand vector $\mathbf{d} = (1, 2, 3)$, as explained in Section IV-A, to apply piggyback coding, the server generates a 3-level Gaussian superposition codebook as follows:
  - $\mathcal{C}_1$ with $\lfloor 2^{nR_3} \rfloor$ rows and $\lfloor 2^{n(R_1+2R_2)} \rfloor$ columns,
  - $\mathcal{C}_2$ with $\lfloor 2^{nR_3} \rfloor$ rows and $\lfloor 2^{n(R_1+R_2-R_3)} \rfloor$ columns,
  - $\mathcal{C}_3$ with $\lfloor 2^{nR_3} \rfloor$ rows and $\lfloor 2^{n(R_1-R_3)} \rfloor$ columns,

  which contain i.i.d. codewords of length n generated from zero-mean Gaussian distributions with variances P_1, P_2, and P_3, respectively.

- Encoding at server: The server transmits

$$X^n(\mathbf{W}, \mathbf{d})$$
$$= x_1^n(V_{1,\mathbf{d}}^r, V_{1,\mathbf{d}}^c) + x_2^n(V_{2,\mathbf{d}}^r, V_{2,\mathbf{d}}^c) + x_3^n(V_{3,\mathbf{d}}^r, V_{3,\mathbf{d}}^c),$$

where

$$V_{1,\mathbf{d}}^r = Z_1, \quad V_{1,\mathbf{d}}^c = (\overline{W}_{\{1\}}, \overline{W}_{\{12\}}, \overline{W}_{\{13\}}),$$
$$V_{2,\mathbf{d}}^r = Z_2, \quad V_{2,\mathbf{d}}^c = (\overline{W}_{\{23\}}^U, \overline{W}_{\{2\}}),$$
$$V_{3,\mathbf{d}}^r = 1, \quad V_{3,\mathbf{d}}^c = \overline{W}_{\{3\}}^U.$$

- Decoding at users:
  - User 1 has the weakest channel gain and needs to receive all the subfiles it has not prefetched, i.e., $\left\{ \overline{W}_{\{1\}}, \overline{W}_{\{12\}}, \overline{W}_{\{13\}} \right\}$. Using its cached content $\overline{W}_{\{123\}}$, it can extract the subcodebook $\left\{ x_1^n(\overline{W}_{\{123\}}, v_1^c) : v_1^c \in [2^{n(R_1+2R_2)}] \right\}$ from $\mathcal{C}_1$ and losslessly recovers the required parts if

$$R_1 + 2R_2 \leq C\left(\frac{h_1^2 P_1}{1 + h_1^2(P_2 + P_3)} \right). \tag{18}$$

  - User 2 requires $\{ \overline{W}_{\{2\}}, \overline{W}_{\{12\}}, \overline{W}_{\{23\}}, \overline{W}_{\{123\}} \}$, and if

$$R_1 + 2R_2 + R_3 \leq C\left(\frac{h_2^2 P_1}{1 + h_2^2(P_2 + P_3)} \right), \tag{19}$$

it can first decode $\overline{W}_{\{123\}}$, $\overline{W}_{\{12\}}$ and $\overline{W}_{\{13\}}$ from the codebook x_1^n, and can retrieve $\overline{W}_{\{23\}}^C$ from its cached contents. It can then decode the remaining parts required to reconstruct file W_2, i.e., parts $\overline{W}_{\{2\}}$ and $\overline{W}_{\{23\}}^U$ from x_2^n using its side information Z_2 if

$$R_1 + R_2 - R_3 \leq C\left(\frac{h_2^2 P_2}{1 + h_2^2 P_3} \right). \tag{20}$$

– User 3 can decode messages $\{\overline{W}_{\{1\}}, \overline{W}_{\{12\}}, \overline{W}_{\{13\}}, \overline{W}_{\{123\}}\}$ from x_1^n if (19) is satisfied, since $h_3 \geq h_2$, and decode messages $\{\overline{W}_{\{2\}}, \overline{W}_{\{23\}}^U, \overline{W}_{\{12\}}^C \oplus \overline{W}_{\{13\}}^C \oplus \overline{W}_{\{23\}}^C\}$ from x_2^n, if

$$R_1 + R_2 \leq C\left(\frac{h_2^2 P_2}{1 + h_2^2 P_3}\right). \qquad (21)$$

With $\overline{W}_{\{12\}}, \overline{W}_{\{13\}}$, it can decode $\overline{W}_{23}^C$ using the coded side information in its cache. Then, only subfile $\overline{W}_{\{3\}}$ is left for user 3 to fully recover W_3. To this end, it can recover $\overline{W}_{\{3\}}^C$ from its cache as it has already decoded $\overline{W}_1^c$ and $\overline{W}_2^c$. Finally, it can decode $\overline{W}_{\{3\}}^U$ from x_3^n if

$$R_1 - R_3 \leq C\left(h_3^2 P_3\right). \qquad (22)$$

The transmission powers P_1, P_2, P_3 are chosen to satisfy Eqs (18)-(22). As it can been seen from the example, the idea is to jointly encode the cached contents of each user together with the message intended for it. This additional message does not interfere with the weak user as it already has it cached, while the stronger users can recover this information without any additional transmission cost.

C. Proposed Scheme

We now present the proposed coded caching and joint encoding scheme for a general setting with $N \geq K$, and a normalized cache capacity $M \leq \min\{R_{N-K+1}, \ldots, R_N\}$. We will explain later how the scheme can be applied to arbitrary number of users and files.

1) Placement Phase: Each subfile $\overline{W}_{\mathcal{S}}, \mathcal{S} \subseteq [N]$, is divided into two non-overlapping parts, $\overline{W}_{\mathcal{S}} = (\overline{W}_{\mathcal{S}}^C, \overline{W}_{\mathcal{S}}^U)$, where $\overline{W}_{\mathcal{S}}^C$ is at rate M, and $\overline{W}_{\mathcal{S}}^U$ is at rate $R_{|\mathcal{S}|} - M$. User $k \in [K]$ caches a linear combination of all the parts $\overline{W}_{\mathcal{S}}^C$ in sublibrary L_{N-k+1} as

$$Z_k = \bigoplus_{\mathcal{S} \subseteq [N]: |\mathcal{S}| = N-k+1} \overline{W}_{\mathcal{S}}^C, \qquad (23)$$

which satisfies the cache capacity constraint M.

2) Delivery Phase: For any demand vector $\mathbf{d} = (d_1, \ldots, d_K) \in [N]^K$, let $N_e(\mathbf{d})$ denote the number of distinct requests in demand $\mathbf{d}$, and let $\mathcal{U} \triangleq \{k_1, \ldots, k_{N_e(\mathbf{d})}\}$ denote the set of users with the weakest channels that request distinct files such that $|\mathcal{U}| = N_e(\mathbf{d})$, where $k_1 < \cdots < k_{N_e(\mathbf{d})}$.

• Codebook construction: The server constructs a $N_e(\mathbf{d})$-level Gaussian superposition codebook, such that for $i \in [N_e]$, the ith-level codebook contains $2^{n\rho_i}$ codewords, where $\rho_i = \sum_{\ell=1}^{N-i+1} \binom{N-i+1}{\ell-1} R_\ell$. If $k_i = i$, the codewords are arranged in an array of 2^{nM} rows and $2^{n(\rho_i - M)}$ columns; otherwise, i.e., $k_i \neq i$, they are arranged into 1 row and $2^{n\rho_i}$ columns. For each element of the array we generate an i.i.d. codeword $x_i^n(v_i^r, v_i^c)$, $v_i^r \in [2^{nM}]$ and $v_i^c \in [2^{n(\rho_i - M)}]$ if $k_i = i$; $v_i^r = 1$ and $v_i^c \in [2^{n\rho_i}]$ if $k_i \neq i$, with distribution $\mathcal{N}(0, P_i)$.

• Encoding at server: The server transmits codeword $\sum_{i=1}^{N_e(\mathbf{d})} x_i^n(V_{i,\mathbf{d}}^r, V_{i,\mathbf{d}}^c)$, where, for $i \in [N_e(\mathbf{d})]$, message

$$V_{i,\mathbf{d}}^r = \begin{cases} Z_{k_i}, & \text{if } k_i = i, \\ \emptyset, & \text{if } k_i \neq i, \end{cases} \qquad (24)$$

is targeted at users $k_i + 1, \ldots, K$, and message

$$V_{i,\mathbf{d}}^c = \begin{cases} \overline{W}_{\widetilde{\mathcal{S}}}^U \cup \left\{\overline{W}_{\mathcal{S}} \in L_{N-i+1} : \mathcal{S} \neq \widetilde{\mathcal{S}}\right\} \\ \quad \cup \left\{\overline{W}_{\mathcal{S}} \notin L_{N-i+1} : \mathcal{S} \in \mathcal{D}_i\right\} & \text{if } k_i = i, \\ \left\{\overline{W}_{\mathcal{S}} : \mathcal{S} \in \mathcal{D}_i\right\} & \text{if } k_i \neq i, \end{cases} \qquad (25)$$

for any $\widetilde{\mathcal{S}}$ such that $\overline{W}_{\widetilde{\mathcal{S}}} \in L_{N-i+1}$, is targeted at users $k_i, \ldots, K$, where

$$\mathcal{D}_i \triangleq \left\{\mathcal{S} : \mathcal{S} \subseteq [N] \setminus \{d_{k_1}, \ldots, d_{k_{i-1}}\}, d_i \in \mathcal{S}, \right.$$
$$\left. |\mathcal{S}| \leq N - i + 1\right\}$$

is the set of subfiles required to reconstruct file $W_{d_{k_i}}$ requested by user k_i, but not common to any of the files requested by the weaker users, i.e., $W_{d_1}, \ldots, W_{d_{k_{i-1}}}$. Codeword $x_i^n(V_{i,\mathbf{d}}^r, V_{i,\mathbf{d}}^c)$ is generated with average power P_i such that

$$|V_{i,\mathbf{d}}^c| + |V_{i,\mathbf{d}}^r| \leq C\left(\frac{h_{k_i+1}^2 P_i}{1 + h_{k_i+1}^2 \sum_{j=i+1}^{N_e(\mathbf{d})} P_j}\right), \qquad (26)$$

$$|V_{i,\mathbf{d}}^c| \leq C\left(\frac{h_{k_i}^2 P_i}{1 + h_{k_i}^2 \sum_{j=i+1}^{N_e(\mathbf{d})} P_j}\right), \qquad (27)$$

where $|V_{i,\mathbf{d}}^c|$ and $|V_{i,\mathbf{d}}^r|$ denote the rates of $V_{i,\mathbf{d}}^c$ and $V_{i,\mathbf{d}}^r$, respectively.

• Decoding at users:

– For $i \in [N_e(\mathbf{d})]$, user k_i decodes all its desired messages in two steps.

Step 1: In the first step, user k_i recovers all the messages $\{V_{i',\mathbf{d}}^r, V_{i',\mathbf{d}}^c : i' \in [i-1]\}$, which correspond to all the subfiles required to reconstruct files $W_{d_{k_1}}, \ldots, W_{d_{k_{i-1}}}$, by decoding the first $i-1$ level codewords. This can be done with arbitrarily low error probability since condition (26) is satisfied.

Step 2: We note that $V_{i,\mathbf{d}}^r$ is either in user k_i's local cache or is an empty message. Thus, user k_i always has the knowledge of $V_{i,\mathbf{d}}^r$, which together with (27) is satisfied, it can allows the user to successfully decode $V_{i,\mathbf{d}}^c$.

Overall, user k_i recovers the subfiles $\{\overline{W}_{\mathcal{S}} : \mathcal{S} \subseteq [N], \mathcal{S} \cap \{d_1, \ldots, d_{k-1}\} \neq \emptyset, d_k \in \mathcal{S}\}$ in the first step, and the subfiles $\{\overline{W}_{\mathcal{S}} : \mathcal{S} \subseteq [N] \setminus \{d_1, \ldots, d_{k-1}\}, d_k \in \mathcal{S}\}$ in the second step, from which it can fully reconstruct W_{d_k}.

– If $k \notin \mathcal{U}$, then user k has requested the same file as a weaker user $k_i \in \mathcal{U}$, i.e., $k_i \leq k$. Therefore, user k can decode all the messages targeted at user k_i, and since (27) is satisfied, user k can also recover $V_{i,\mathbf{d}}^r$, from which it can fully reconstruct W_{d_k}.

Remark 2: We consider more files than users, i.e., $N \geq K$, but the analysis for case $N < K$ follows directly. Note that, since each user stores a coded combination of all the subfiles in a sublibrary, with more users than files, i.e., $N < K$, the $K - N$ strongest users would be able to decode all of their required subfiles from the messages targeted at users $1, \ldots, N$, rendering the cached contents $Z_{N+1}, \ldots, Z_K$ unutilized.

For any demand vector $\mathbf{d}$, the total transmit power required by the proposed caching scheme can be upper bounded as in the following theorem.

Theorem 2: For the caching problem described in Section II, with cache capacity

$$M \leq \min\left\{ R_\zeta, \ldots, R_N \right\}, \quad \zeta \triangleq \max\{N - K, 1\}$$

an upper bound on the optimal memory-power function, $P^*(M)$, is given by

$$P^*(M) \leq P_{UB}^{PB}(M) \triangleq \sum_{k=1}^{K} P_k(M),$$

where

$$P_k(M) = \begin{cases} 0, & \text{if } k \notin [\min\{N, K\}] \\ \max\left\{ \left(\dfrac{2^{2\tilde{\rho}_k} - 1}{h_k^2} \right)\left(1 + h_k^2 \displaystyle\sum_{j=k+1}^{K} P_j \right), \right. \\ \left. \left(\dfrac{2^{2(\tilde{\rho}_k + M)} - 1}{h_{k+1}^2} \right)\left(1 + h_{k+1}^2 \displaystyle\sum_{j=k+1}^{K} P_j \right) \right\}, \\ \quad \text{if } k \in [\min\{N, K\}], \end{cases}$$

$$\tilde{\rho}_k \triangleq \max\left\{ \sum_{\ell=0}^{N-k} \binom{N-k}{\ell} R_{\ell+1} - M, \; 0 \right\}, \forall k \in [K]. \tag{28}$$

Proof: The proof is given in Appendix B, which is derived by characterizing the transmit power achieved by the caching and delivery scheme described in Section IV-C. □

V. LOWER BOUND

This section provides a lower bound on the memory-power function when the placement phase is limited to caching functions that store uncoded contents.

Theorem 3: For the caching problem described in Section II with uncoded cache placement phase, the optimal memory-power function, $P^*(M)$, is lower bounded as

$$P^*(M) \geq P_{LB}(M) \triangleq \sum_{k=1}^{\min\{N, K\}} \left(\frac{2^{2\tilde{\rho}_k} - 1}{h_k^2} \right) \prod_{j=1}^{k-1} 2^{2\tilde{\rho}_j}, \tag{29}$$

with $\tilde{\rho}_k$ defined as in (28).

We denote by $\mathfrak{D}_d$ the set of all demand combinations such that the first N_e users request distinct files, where $N_e \triangleq \min\{N, K\}$. We note that there are a total of $\binom{N}{N_e} N_e! N^{K-N_e}$ such demand combinations, i.e, $|\mathfrak{D}_d| = \binom{N}{N_e} N_e! N^{K-N_e}$, enumerated as $\mathbf{d}_t \triangleq (d_1^t, \ldots, d_K^t) \in \mathfrak{D}_d$, $t \in \left[\binom{N}{N_e} N_e! N^{K-N_e} \right]$.

Example 3: Consider $N = 3$, $K = 4$. We have $|\mathfrak{D}_d| = 18$ and

$$\begin{aligned} \mathbf{d}_1 &= \{1,2,3,1\}, \quad \mathbf{d}_2 = \{1,2,3,2\}, \quad \mathbf{d}_3 = \{1,2,3,3\}, \\ \mathbf{d}_4 &= \{1,3,2,1\}, \quad \mathbf{d}_5 = \{1,3,2,2\}, \quad \mathbf{d}_6 = \{1,3,2,3\}, \\ \mathbf{d}_7 &= \{2,3,1,1\}, \quad \mathbf{d}_8 = \{2,3,1,2\}, \quad \mathbf{d}_9 = \{2,3,1,3\}, \\ \mathbf{d}_{10} &= \{3,2,1,1\}, \quad \mathbf{d}_{11} = \{3,2,1,2\}, \quad \mathbf{d}_{12} = \{3,2,1,3\}, \\ \mathbf{d}_{13} &= \{3,1,2,1\}, \quad \mathbf{d}_{14} = \{3,1,2,2\}, \quad \mathbf{d}_{15} = \{3,1,2,3\}, \\ \mathbf{d}_{16} &= \{2,1,3,1\}, \quad \mathbf{d}_{17} = \{2,1,3,2\}, \quad \mathbf{d}_{18} = \{2,1,3,3\}. \end{aligned} \tag{30}$$

To prove Theorem 3, we first present the following lemma.

Lemma 1: There exist random variables $X_{\mathbf{d}_t}$, $Y_{1,\mathbf{d}_t}$, …, $Y_{N_e,\mathbf{d}_t}$, where for $X_{\mathbf{d}_t} = x$, $x \in \mathbb{R}$,

$$Y_{k,\mathbf{d}_t} | x \sim \mathcal{N}(h_k x, 1), \quad k \in [N_e], \tag{31}$$

and auxiliary random variables $U_{1,\mathbf{d}_t}$, …, $U_{N_e-1,\mathbf{d}_t}$, such that

$$U_{1,\mathbf{d}_t} - \cdots - U_{N_e-1,\mathbf{d}_t} - X_{\mathbf{d}_t} - Y_{N_e,\mathbf{d}_t} - \cdots - Y_{1,\mathbf{d}_t} \tag{32}$$

form a Markov chain, and

$$\begin{aligned} &\frac{1}{n} H(W_{d_1^t}) + \epsilon_n \\ &\quad \leq \frac{1}{n} I(W_{d_1^t}; Z_1) + I(U_{1,\mathbf{d}_t}; Y_{1,\mathbf{d}_t}); \\ &\frac{1}{n} H(W_{d_k^t} | W_{d_{k-1}^t}, \ldots, W_{d_1^t}) + \epsilon_n \\ &\quad \leq \frac{1}{n} I(W_{d_k^t}; Z_1, \ldots, Z_k | W_{d_{k-1}^t}, \ldots, W_{d_1^t}) \\ &\quad\quad + I(U_{k,\mathbf{d}_t}; Y_{k,\mathbf{d}_t} | U_{k-1,\mathbf{d}_t}), k \in [2 : N_e - 1]; \\ &\frac{1}{n} H(W_{d_{N_e}^t} | W_{d_{N_e-1}^t}, \ldots, W_{d_1^t}) + \epsilon_n \\ &\quad \leq \frac{1}{n} I(W_{d_{N_e}^t}; Z_1, \ldots, Z_{N_e} | W_{d_{N_e-1}^t}, \ldots, W_{d_1^t}) \\ &\quad\quad + I(X_{\mathbf{d}_t}; Y_{N_e,\mathbf{d}_t} | U_{N_e-1,\mathbf{d}_t}), \end{aligned} \tag{33}$$

where ϵ_n goes to zero as $n \to \infty$.

Proof: The proof of the above lemma is similar to the proof of [14, Lemma 14], which we omit here. □

Proof: For any demand vector $\mathbf{d}_t \triangleq (d_1^t, \ldots, d_K^t) \in \mathfrak{D}_d$, we have $H(W_{d_1^t}) = nR$, and

$$H(W_{d_k^t} | W_{d_{k-1}^t}, \ldots, W_{d_1^t})$$

$$= H\left(\bigcup_{\substack{\mathcal{S} \subseteq [N] \\ \mathcal{S} \ni d_k^t}} \overline{W}_{\mathcal{S}} \;\middle|\; \bigcup_{\substack{\mathcal{S} \subset [N] \\ \{d_{k-1}^t, \ldots, d_1^t\} \cap \mathcal{S} \neq \emptyset}} \overline{W}_{\mathcal{S}} \right) \tag{34a}$$

$$= H\left(\bigcup_{\substack{\mathcal{S} \subseteq [N] \setminus \{d_{k-1}^t, \ldots, d_1^t\} \\ \mathcal{S} \ni d_k^t}} \overline{W}_{\mathcal{S}} \right)$$

$$= \sum_{\ell=0}^{N-k} \binom{N-k}{\ell} n R_{\ell+1}, \quad k \in \{2, \ldots, N_e\}, \tag{34b}$$

where (34a) follows from the fact that $W_i = \bigcup_{\substack{\mathcal{S} \subseteq [N] \\ \mathcal{S} \ni i}} \overline{W}_\mathcal{S}$, $\forall i \in [N]$, and (34b) follows due to the independence of the subfiles. Similarly, we have $I(W_{d_1^t}; Z_1) = I\left(\bigcup_{\substack{\mathcal{S} \subseteq [N] \\ \mathcal{S} \ni d_1^t}} \overline{W}_\mathcal{S}; Z_1 \right)$, and

$$I(W_{d_k^t}; Z_1, \ldots, Z_k | W_{d_{k-1}^t}, \ldots, W_{d_1^t})$$

$$= I\left(\bigcup_{\substack{\mathcal{S} \subseteq [N] \\ \mathcal{S} \ni d_k^t}} \overline{W}_\mathcal{S}; Z_1, \ldots, Z_k \middle| \bigcup_{\substack{\mathcal{S} \subseteq [N] \\ \{d_{k-1}^t, \ldots, d_1^t\} \cap \mathcal{S} \neq \emptyset}} \overline{W}_\mathcal{S} \right) \quad (35a)$$

$$= I\left(\bigcup_{\substack{\mathcal{S} \subseteq [N] \backslash \{d_{k-1}^t, \ldots, d_1^t\} \\ \mathcal{S} \ni d_k^t}} \overline{W}_\mathcal{S}; Z_1, \ldots, Z_k \middle| \bigcup_{\substack{\mathcal{S} \subseteq [N] \\ \{d_{k-1}^t, \ldots, d_1^t\} \cap \mathcal{S} \neq \emptyset}} \overline{W}_\mathcal{S} \right) \quad (35b)$$

$$= H\left(\bigcup_{\substack{\mathcal{S} \subseteq [N] \backslash \{d_{k-1}^t, \ldots, d_1^t\} \\ \mathcal{S} \ni d_k^t}} \overline{W}_\mathcal{S} \middle| \bigcup_{\substack{\mathcal{S} \subseteq [N] \\ \{d_{k-1}^t, \ldots, d_1^t\} \cap \mathcal{S} \neq \emptyset}} \overline{W}_\mathcal{S} \right)$$

$$- H\left(\bigcup_{\substack{\mathcal{S} \subseteq [N] \backslash \{d_{k-1}^t, \ldots, d_1^t\} \\ \mathcal{S} \ni d_k^t}} \overline{W}_\mathcal{S} \middle| Z_1, \ldots, Z_k \right)$$

$$+ I\left(\bigcup_{\substack{\mathcal{S} \subseteq [N] \backslash \{d_{k-1}^t, \ldots, d_1^t\} \\ \mathcal{S} \ni d_k^t}} \overline{W}_\mathcal{S}; \bigcup_{\substack{\mathcal{S} \subseteq [N] \\ \{d_{k-1}^t, \ldots, d_1^t\} \cap \mathcal{S} \neq \emptyset}} \overline{W}_\mathcal{S} \middle| Z_1, \ldots, Z_k \right) \quad (35c)$$

$$= H\left(\bigcup_{\substack{\mathcal{S} \subseteq [N] \backslash \{d_{k-1}^t, \ldots, d_1^t\} \\ \mathcal{S} \ni d_k^t}} \overline{W}_\mathcal{S} \right)$$

$$- H\left(\bigcup_{\substack{\mathcal{S} \subseteq [N] \backslash \{d_{k-1}^t, \ldots, d_1^t\} \\ \mathcal{S} \ni d_k^t}} \overline{W}_\mathcal{S} \middle| Z_1, \ldots, Z_k \right) \quad (35d)$$

$$\leq I\left(\bigcup_{\substack{\mathcal{S} \subseteq [N] \backslash \{d_{k-1}^t, \ldots, d_1^t\} \\ \mathcal{S} \ni d_k^t}} \overline{W}_\mathcal{S}; Z_1, \ldots, Z_k \right), \quad (35e)$$

for $k \in [2 : N_e]$, where (35d) follows due to the independence of the subfiles and uncoded cache placement. In particular, the first term in (35c) equals to the first term in (35d) since different subfiles are independent. And also because Z_k, $\forall k \in \{1, \ldots, K\}$ contains only uncoded contents from each subfile, the last term in (35c) equals to 0. Thus, for n sufficiently large,

we can rewrite (33) as

$$R \leq \frac{1}{n} I\left(\bigcup_{\substack{\mathcal{S} \subseteq [N] \\ \mathcal{S} \ni d_k^1}} \overline{W}_\mathcal{S}; Z_1 \right) + I(U_{1,\mathbf{d}_t}; Y_{1,\mathbf{d}_t});$$

$$\sum_{\ell=0}^{N-k} \binom{N-k}{l} R_{\ell+1}$$

$$\leq \frac{1}{n} I\left(\bigcup_{\substack{\mathcal{S} \subseteq [N] \backslash \{d_{k-1}^t, \ldots, d_1^t\} \\ \mathcal{S} \ni d_k^t}} \overline{W}_\mathcal{S}; Z_1, \ldots, Z_k \right)$$

$$+ I(U_{k,\mathbf{d}_t}; Y_{k,\mathbf{d}_t} | U_{k-1,\mathbf{d}_t}), k \in [2 : N_e - 1];$$

$$\sum_{l=0}^{N-N_e} \binom{N-N_e}{\ell} R_{\ell+1}$$

$$\leq \frac{1}{n} I\left(\bigcup_{\substack{\mathcal{S} \subseteq [N] \backslash \{d_{N_e-1}^t, \ldots, d_1^t\} \\ \mathcal{S} \ni d_{N_e}^t}} \overline{W}_\mathcal{S}; Z_1, \ldots, Z_{N_e} \right)$$

$$+ I(X_{\mathbf{d}_t}; Y_{N_e,\mathbf{d}_t} | U_{N_e-1,\mathbf{d}_t}). \quad (36)$$

Let $\tau_k \triangleq I(U_{k,\mathbf{d}_t}; Y_{k,\mathbf{d}_t} | U_{k-1,\mathbf{d}_t})$, for $k = 1, \ldots, N_e$, where we set $U_{0,\mathbf{d}_t} \triangleq 0$ and $U_{N_e,\mathbf{d}_t} \triangleq X_{\mathbf{d}_t}$. Since $X_{\mathbf{d}_t}, Y_{1,\mathbf{d}_t}, \ldots, Y_{N_e,\mathbf{d}_t}, U_{1,\mathbf{d}_t}, \ldots, U_{N_e-1,\mathbf{d}_t}$ satisfy (31) and (32), rate-tuple $(\tau_1, \ldots, \tau_{N_e})$ is within the capacity region of the underlying degraded Gaussian BC described in Section II [14], [24]. According to the capacity region of degraded Gaussian BC characterized in [23], we have

$$c_k(\mathbf{d}_t) \leq \tau_k \leq \frac{1}{2} \log_2 \left(1 + \frac{h_k^2 P_k(\mathbf{d}_t)}{h_k^2 \sum_{j=k+1}^{N_e} P_j(\mathbf{d}_t) + 1} \right),$$

where

$$c_k(\mathbf{d}_t) \triangleq \sum_{\ell=0}^{N-k} \binom{N-k}{\ell} R_{\ell+1}$$

$$- \frac{1}{n} I\left(\bigcup_{\substack{\mathcal{S} \subseteq [N] \backslash \{d_{k-1}^t, \ldots, d_1^t\} \\ \mathcal{S} \ni d_k^t}} \overline{W}_\mathcal{S}; Z_1, \ldots, Z_k \right),$$

$$k \in [N_e]. \quad (37)$$

Thus, with (36) and according to Proposition 1, the required average transmission power to satisfy any demand vector $\mathbf{d}_t \in \mathfrak{D}_d$ is lower bounded by

$$P(\mathbf{d}_t) \geq \sum_{k=1}^{N_e} P_k(\mathbf{d}_t) = q(c_1(\mathbf{d}_t), \ldots, c_{N_e}(\mathbf{d}_t)),$$

where

$$q(c_1(\mathbf{d}_t), \ldots, c_{N_e}(\mathbf{d}_t)) \triangleq \sum_{k=1}^{N_e} \left(\frac{2^{2c_k(\mathbf{d}_t)} - 1}{h_k^2} \right) \prod_{j=1}^{k-1} 2^{2c_j(\mathbf{d}_t)}; \quad (38)$$

It is proved in [12, Appendix B] that $q(\cdot)$ is a convex function of $(c_1(\mathbf{d}_t), \ldots, c_{N_e}(\mathbf{d}_t))$. Thus, the optimal achievable power is lower bounded by

$$P^*(M) \geq \frac{1}{|\mathfrak{D}_d|} \sum_{t=1}^{|\mathfrak{D}_d|} P(\mathbf{d}_t) \geq \frac{1}{|\mathfrak{D}_d|} \sum_{t=1}^{|\mathfrak{D}_d|} q(c_1(\mathbf{d}_t), \ldots, c_{N_e}(\mathbf{d}_t)) \tag{39a}$$

$$\geq q\left(\frac{1}{|\mathfrak{D}_d|} \sum_{t=1}^{|\mathfrak{D}_d|} c_1(\mathbf{d}_t), \ldots, \frac{1}{|\mathfrak{D}_d|} \sum_{t=1}^{|\mathfrak{D}_d|} c_{N_e}(\mathbf{d}_t) \right)$$

$$\geq \sum_{k=1}^{N_e} \left(\frac{2^{2\tilde{\rho}_k} - 1}{h_k^2} \right) \prod_{j=1}^{k-1} 2^{2\tilde{\rho}_j}, \tag{39b}$$

where we recall that

$$\tilde{\rho}_k \triangleq \max\left\{ \sum_{\ell=0}^{N-k} \binom{N-k}{\ell} R_{\ell+1} - M, 0 \right\}.$$

(39b) follows from the convexity of $q(\cdot)$, and (39b) holds since $\frac{1}{|\mathfrak{D}_d|} \sum_{t=1}^{|\mathfrak{D}_d|} c_k(\mathbf{d}_t) \geq \tilde{\rho}_k, \forall k \in [N_e]$, which we will prove in the following. For any $k \in [N_e]$, divide all the demands $\mathbf{d}_t \in \mathfrak{D}_d$ into $|\mathfrak{D}_d|/k$ disjoint groups, where each has k demand vectors such that $d_k^{t_1} \in \{d_1^{t_2}, \ldots, d_{k-1}^{t_2}\}$, and $d_k^{t_2} \in \{d_1^{t_1}, \ldots, d_{k-1}^{t_1}\}$, if $\mathbf{d}_{t_1}$ and $\mathbf{d}_{t_2}$ are in the same group, and $t_1 \neq t_2$.

In Example 3, there are 18 vectors in $\mathfrak{D}_d$ listed in (30). For $k = 3$, one partition that meets the above condition is

$$G_1 = \{\mathbf{d}_1, \mathbf{d}_4, \mathbf{d}_7\}, \quad G_2 = \{\mathbf{d}_2, \mathbf{d}_5, \mathbf{d}_8\},$$
$$G_3 = \{\mathbf{d}_3, \mathbf{d}_6, \mathbf{d}_9\}, \quad G_4 = \{\mathbf{d}_{10}, \mathbf{d}_{13}, \mathbf{d}_{16}\},$$
$$G_5 = \{\mathbf{d}_{11}, \mathbf{d}_{14}, \mathbf{d}_{17}\}, \quad G_6 = \{\mathbf{d}_{12}, \mathbf{d}_{15}, \mathbf{d}_{18}\},$$

where G_j, $j \in [6]$, denotes one group that satisfies $d_3^{t_1} \in \{d_1^{t_2}, d_2^{t_2}\}$, and $d_3^{t_2} \in \{d_1^{t_1}, d_2^{t_1}\}$, $\forall \mathbf{d}_{t_1}, \mathbf{d}_{t_2} \in G_j, t_1 \neq t_2$.

We denote the index of the sth demand vector in the jth group by t_{js}. Thus,

$$\frac{1}{|\mathfrak{D}_d|} \sum_{t=1}^{|\mathfrak{D}_d|} \frac{1}{n} I\left(\bigcup_{\substack{\mathcal{S} \subseteq [N] \setminus \{d_{k-1}^t, \ldots, d_1^t\} \\ \mathcal{S} \ni d_k^t}} \overline{W}_{\mathcal{S}}; Z_1, \ldots, Z_k \right)$$

$$= \frac{1}{|\mathfrak{D}_d|} \sum_{j=1}^{|\mathfrak{D}_d|/k} \sum_{s=1}^{k} \frac{1}{n} I\left(\bigcup_{\substack{\mathcal{S} \subseteq [N] \setminus \{d_{k-1}^{t_{js}}, \ldots, d_1^{t_{js}}\} \\ \mathcal{S} \ni d_k^{t_{js}}}} \overline{W}_{\mathcal{S}}; Z_1, \ldots, Z_k \right) \tag{40a}$$

$$= \frac{1}{|\mathfrak{D}_d|} \sum_{j=1}^{|\mathfrak{D}_d|/k} \frac{1}{n} I\left(\bigcup_{s \in [k]} \bigcup_{\substack{\mathcal{S} \subseteq [N] \setminus \{d_{k-1}^{t_{js}}, \ldots, d_1^{t_{js}}\} \\ \mathcal{S} \ni d_k^{t_{js}}}} \overline{W}_{\mathcal{S}}; Z_1, \ldots, Z_k \right) \tag{40b}$$

$$\leq \frac{1}{|\mathfrak{D}_d|} \sum_{j=1}^{|\mathfrak{D}_d|/k} \frac{1}{n} \min\left\{ H\left(\bigcup_{s \in [k]} \bigcup_{\substack{\mathcal{S} \subset [N] \setminus \{d_{k-1}^{t_{js}}, \ldots, d_1^{t_{js}}\} \\ d_k^{t_{js}} \in \mathcal{S}}} \overline{W}_{\mathcal{S}} \right), \right.$$
$$\left. H(Z_1, \ldots, Z_k) \right\} \tag{40c}$$

$$= \min\left\{ \sum_{\ell=0}^{N-k} \binom{N-k}{\ell} R_{\ell+1}, M \right\}, \tag{40d}$$

where (40a) is derived by writing the summation with regards to the groups; (40b) follows the independence of subfiles and the fact that

$$\bigcap_{s \in [k]} \left(\bigcup_{\substack{\mathcal{S} \subset [N] \setminus \{d_{k-1}^{t_{js}}, \ldots, d_1^{t_{js}}\} \\ d_k^{t_{js}} \in \mathcal{S}}} \overline{W}_{\mathcal{S}} \right) = \emptyset,$$

since $d_k^{t_{js_1}} \in \{d_1^{t_{js_2}}, \ldots, d_{k-1}^{t_{js_2}}\}$, while $d_k^{t_{js_2}} \in \{d_1^{t_{js_1}}, \ldots, d_{k-1}^{t_{js_1}}\}$, if $s_1 \neq s_2$, $\forall s_1, s_2 \in [k], j \in [|\mathfrak{D}_d|/k]$; (40c) follows since mutual information is no larger than the entropy of each component. (40d) follows from the size of the subfiles and the cache capacity. Substituting (37) and (40d) into (39b), we have proven (29). Thus, the proof of Theorem 3 is completed. □

Remark 3: We observe that, if

$$\left(\frac{2^{2\tilde{\rho}_k} - 1}{h_k^2} \right) \left(1 + h_k^2 \sum_{j=k+1}^{K} P_j \right)$$

$$\geq \left(\frac{2^{2(\tilde{\rho}_k + M)} - 1}{h_{k+1}^2} \right) \left(1 + h_{k+1}^2 \sum_{j=k+1}^{K} P_j \right),$$
$$\forall k \in [\min\{N, K\}], \tag{41}$$

then $P_{UB}^{PB}(M) = P_{LB}(M)$, i.e., the transmission power required by the coded placement and joint encoding scheme meets the lower bound. However, it does not necessarily mean that this scheme is optimal as the lower bound is derived assuming uncoded placement phase, while the proposed scheme caches contents in a coded manner. Nevertheless, we can conclude that the performance of the proposed scheme is no worse than the optimal scheme with uncoded placement phase.

VI. NUMERICAL RESULTS

We evaluate the performance of the scheme proposed in Sec III-B, referred to as the *correlation-aware* scheme, by comparing its memory-power trade-off with the lower bound presented in Theorem 3, as well as with the trade-off achieved by the scheme proposed in [12], which does not exploit the correlation among files, referred to as the *correlation-ignorant* scheme. In the latter scheme, we treat
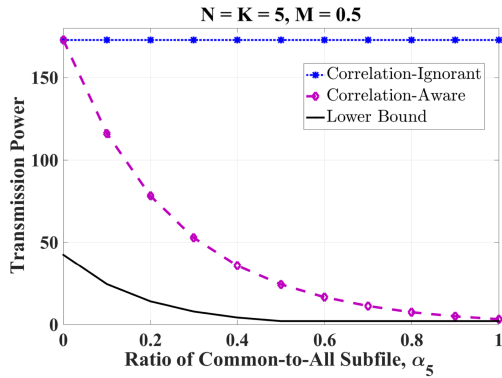
Fig. 2. Transmission power vs. common subfile fraction, when the files are composed of private and common-to-all subfiles. The channel gains are given as $1/h_k^2 = 2 - 0.2(k - 1)$, $k = 1, \ldots, 5$. The correlation-aware scheme corresponds to the superposition coding scheme in Section III-B.
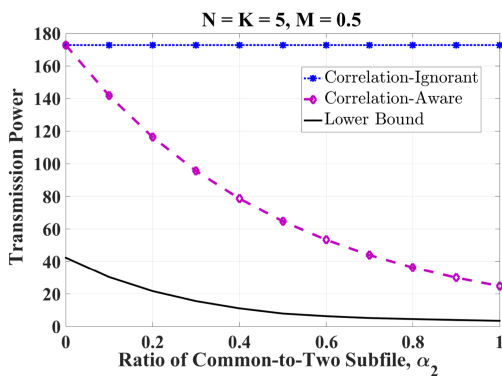

Fig. 3. Transmission power vs. common subfile fraction, when the files are composed of private and common-to-two subfiles. The channel gains are given as $1/h_k^2 = 2 - 0.2(k - 1)$, $k = 1, \ldots, 5$. The correlation-aware scheme corresponds to the superposition coding scheme in Section III-B.

each file as a distinct sequence of bits. We consider a setting with $N = 5$ files, $K = 5$ users, file rate $R = 1$, and cache capacity $M = 0.5$. Channel gains are modeled as $1/h_k^2 = 2 - 0.2(k - 1)$, for $k = 1, \ldots, 5$. We denote by α_ℓ the file-length fraction that belongs to sublibrary L_ℓ, i.e.,

$$\alpha_\ell = \binom{N-1}{\ell-1} \frac{R_\ell}{R}, \quad \sum_{\ell=1}^{N} \alpha_\ell = 1.$$

Fig. 2 displays the memory-power trade-off for a database with files composed of one *private* subfile, which is exclusive to that file, and a *common-to-all* subfile, which is shared among all the files, i.e., $\alpha_1 + \alpha_5 = 1$, $\alpha_2 = \alpha_3 = \alpha_4 = 0$. In Fig. 3 the trade-off is shown when the files, in addition to private subfiles, have pairwise correlations through *common-to-two* subfiles, that is $\alpha_1 + \alpha_2 = 1$, $\alpha_3 = \alpha_4 = \alpha_5 = 0$. We plot the minimum transmit power as a function of the common parts of the files for both scenarios, i.e., with respect to α_5 and α_2, respectively. In both settings the transmission power achieved by the correlation-aware scheme decreases remarkably, as the portion of common subfiles increases, while the performance of the correlation-ignorant scheme does not improve. It is observed that the transmission power drops faster in Fig. 2 compared to Fig. 3 for increasing ratio of common
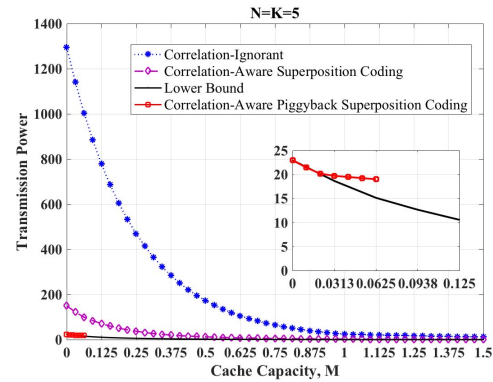

Fig. 4. Transmission power vs. cache capacity, $1/h_k^2 = 2 - 0.2(k - 1)$, for $k = 1, \ldots, K$. The portions of subfiles of different correlation level are specified by $\alpha_1 = \alpha_5 = 1/16$, $\alpha_2 = \alpha_4 = 1/4$, and $\alpha_3 = 3/8$. Correlation-aware superposition coding and piggyback superposition coding correspond to the schemes proposed in Section III-B and Section IV-C, respectively.
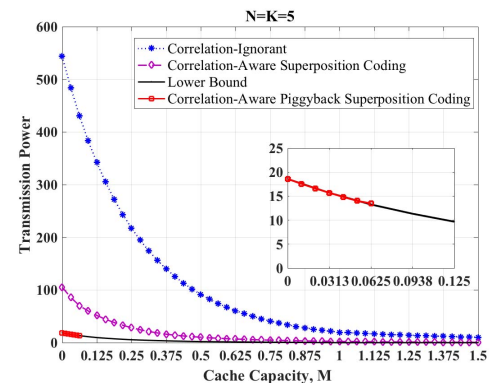

Fig. 5. Transmission power vs. cache capacity, $1/h_k^2 = 2 - 0.4(k - 1)$, for $k = 1, \ldots, K$. The portions of subfiles of different correlation level are specified by $\alpha_1 = \alpha_5 = 1/16$, $\alpha_2 = \alpha_4 = 1/4$, and $\alpha_3 = 3/8$. Correlation-aware superposition coding and piggyback superposition coding correspond to the schemes proposed in Section III-B and Section IV-C, respectively.

subfiles, in both the correlation-aware scheme and the lower bound. This is due to the reduction in the amount of content that needs to be sent over the Gaussian BC for a higher level of correlation among the files. For example, in Fig. 2, as α_5 approaches 1, all the files become the same, and hence, only a message of rate $R/2$ needs to be multicasted to all the users, whereas in the setting of Fig. 3, with $\alpha_2 = 1$, we still have $\binom{N}{2} = 10$ distinct subfiles each shared by only two files. It is also observed that the gap between the transmit power upper and lower bounds is smaller in Fig. 2 compared to Fig. 3.

Next, we consider the same setting with $N = 5$, $K = 5$, and $R = 1$ as before, but let $R_1 = R_2 = \cdots = R_N$, i.e., each subfile has the same size, which yields $\alpha_1 = \alpha_5 = 1/16$, $\alpha_2 = \alpha_4 = 1/4$, and $\alpha_3 = 3/8$. In Fig. 4, the channel gains are given as $1/h_k^2 = 2 - 0.2(k - 1)$, for $k = 1, \ldots, 5$, while in Fig. 5, $1/h_k^2 = 2 - 0.4(k - 1)$, for $k = 1, \ldots, 5$. We compare the proposed scheme presented in Section III, referred to as *Correlation-Aware Superposition Coding*, the joint encoding scheme with coded placement presented in Section IV, referred to *Correlation-Aware Piggyback*

Superpostion Coding, with the correlation ignorant scheme, and the lower bound as well. In both cases, the joint encoding scheme with coded placement can be applied when $M \leq 1/16$. We observe that the correlation-aware schemes significantly outperform the correlation-ignorant scheme in terms of transmission power, and the joint encoding scheme with coded placement further improves the energy efficiency remarkably and achieves approximately the lower bound. However, while in the zoomed-in figure of Fig. 5, the joint scheme meets the lower bound, it can be seen in the zoomed-in figure of Fig. 4 that the joint scheme results in a slightly higher transmission power than the lower bound when the cache capacity is larger than a certain value. That is because the channel of a stronger user is not good enough to receive all the contents (which are the cached contents at the weaker user) piggybacked on the message intended for the weaker user without any additional cost, such that (41) is not satisfied.

VII. CONCLUSION

We have investigated caching and delivery of correlated contents over a K-user Gaussian BC for users with equal-capacity caches. Correlation among files is captured by the component subfiles shared among different subsets of files. We have presented two upper bounds on the memory-power trade-off with correlation-aware cache-aided coding schemes. The first scheme generates coded packets according to user demands, which are then delivered to users using superposition coding, where each coded packet is targeted at the weakest user demanding it. We have also proposed a coded placement scheme with joint encoding, in which the cache contents and user demands are encoded jointly, such that the weak users can use their cache contents for decoding, while the stronger users can decode both without additional resources. We have also derived a lower bound on the minimum transmission power with which any possible demand combination can be satisfied, assuming uncoded cache placement.

Our numerical results indicate that the proposed coding schemes greatly improve the energy-efficiency of delivery over Gaussian BCs compared to correlation-ignorant schemes. For small cache memory sizes, the joint encoding scheme with coded caching requires a lower transmit power, which meets the lower bound assuming uncoded placement under certain conditions. A tighter lower bound without the limitation to uncoded placement is considered as future work.

APPENDIX A
PROOF OF THEOREM 1

To prove Theorem 1, we show the required transmission power by the proposed caching and delivery scheme presented in Section III is upper bounded by $P_{UB}(M, \boldsymbol{\pi})$ for any demand combination $\mathbf{d}$, given cache allocation vector $\boldsymbol{\pi}$.

Recall that for a given demand combination $\mathbf{d} = (d_1, \ldots, d_K)$, $\mathcal{D} = \{d_1, \ldots, d_K\}$. For $\ell \in [N]$, $r \in [\max\{\ell - N + |\mathcal{D}|, 1\} : \min\{\ell, |\mathcal{D}|\}]$, $\mathcal{W}_r = \{\overline{W}_{\mathcal{S}} : |S| = \ell, |\mathcal{S} \cap \mathcal{D}| = r\}$ consists of $\binom{N-|\mathcal{D}|}{\ell - r}\binom{|\mathcal{D}|}{r}$ subfiles. Function GROUP generates $\binom{N-|\mathcal{D}|}{\ell - r}\binom{|\mathcal{D}|-1}{r-1}$ groups based on $\mathcal{W}_r$. For each group $\mathfrak{S}_i = (\mathcal{S}_1, \ldots, \mathcal{S}_K)$, Algorithm 1 runs

function SINGLE-DEMAND twice (code line 6 to 7) to generates two set of coded messages V_1^A, ..., V_K^A and V_1^B, ..., V_K^B corresponding to $\{\overline{W}_{\mathcal{S},\mathcal{A}}^A\}$ and $\{\overline{W}_{\mathcal{S},\mathcal{B}}^B\}$, respectively. We recall that $\mathcal{K}$ is the set of the weakest users with distinct demands according to $\mathfrak{S}_i$, where $\mathcal{K} \triangleq \{k : \mathcal{S}_k \notin \{\mathcal{S}_1, \ldots, \mathcal{S}_{k-1}\}\}$ (line 4 of function SINGLE-DEMAND), and denote by e_k the number of leaders after user k, i.e., $e_k \triangleq \sum_{k'=k+1}^{K} \mathbb{1}\{k' \in \mathcal{K}\}$. Then for $k \in [K]$, the total size of V_k^A and V_k^B denoted by $\hat{\gamma}_{k,\ell,r}(\mathcal{K})$ (normalized by n), i.e., $\hat{\gamma}_{k,\ell,r}(\mathcal{K}) \triangleq |V_k^A| + |V_k^B|$, is given by

$$
\hat{\gamma}_{k,\ell,r}(\mathcal{K}) = \begin{cases} \dfrac{\binom{K-k}{t_\ell^A}}{\binom{K}{t_\ell^A}}(t_\ell^B - t_\ell)R_l + \dfrac{\binom{K-k}{t_\ell^B}}{\binom{K}{t_\ell^B}}(t_\ell - t_\ell^A)R_\ell \\ \quad \text{if } k \in \mathcal{K}, \\ \dfrac{\binom{K-k}{t_\ell^A} - \binom{K-k-e_k}{t_\ell^A}}{\binom{K}{t_\ell^A}}(t_\ell^B - t_\ell)R_\ell \\ \quad + \dfrac{\binom{K-k}{t_\ell^B} - \binom{K-k-e_k}{t_\ell^B}}{\binom{K}{t_\ell^B}}(t_\ell - t_\ell^A)R_\ell \\ \quad \text{if } k \notin \mathcal{K}. \end{cases}
$$
(42)

Thus, the additional power required to send coded messages $V_1^A, \ldots, V_K^A$, and $V_1^B, \ldots, V_K^B$, denoted by ΔP, is given as

$$
\begin{aligned}
&\Delta P(\hat{\gamma}_{1,\ell,r}(\mathcal{K}), \ldots, \hat{\gamma}_{K,\ell,r}(\mathcal{K})) \\
&= \sum_{k=1}^{K}\left(\frac{2^{2(\overline{\rho}_k + \hat{\gamma}_{k,\ell,r}(\mathcal{K}))} - 1}{h_k^2}\right)\prod_{j=1}^{k-1}\frac{2^{2(\overline{\rho}_j + \hat{\gamma}_{j,\ell,r}(\mathcal{K}))}}{h_j^2} \\
&\quad - \sum_{k=1}^{K}\left(\frac{2^{2\overline{\rho}_k} - 1}{h_k^2}\right)\prod_{j=1}^{k-1}\frac{2^{2\overline{\rho}_j}}{h_j^2},
\end{aligned}
$$

where $\overline{\rho}_1, \ldots, \overline{\rho}_1 \in \mathbb{R}^+$ denote the total rate of all the other coded message required to be sent over the Gaussian BC.

Note that

$$
\hat{\gamma}_{k,\ell,r}([|\mathcal{K}|]) \\
= \begin{cases} \dfrac{\binom{K-k}{t_\ell^A}}{\binom{K}{t_\ell^A}}(t_\ell^B - t_\ell)R_\ell + \dfrac{\binom{K-k}{t_\ell^B}}{\binom{K}{t_\ell^B}}(t_\ell - t_\ell^A)R_\ell & \text{if } k \in [|\mathcal{K}|], \\ 0 & \text{if } k \notin [|\mathcal{K}|]. \end{cases}
$$
(43)

Compare (42) and (43). We have then

$$
\begin{aligned}
\sum_{k=1}^{K}\hat{\gamma}_{k,\ell,r}(\mathcal{K}) &= \sum_{k=1}^{K}\hat{\gamma}_{k,\ell,r}([|\mathcal{K}|]) \\
&= \frac{\binom{K}{t_\ell^A+1} - \binom{K-|\mathcal{K}|}{t_\ell^A+1}}{\binom{K}{t_\ell^A}}(t_\ell^B - t_\ell)R_\ell \\
&\quad + \frac{\binom{K}{t_\ell^B+1} - \binom{K-|\mathcal{K}|}{t_\ell^B+1}}{\binom{K}{t_\ell^B}}(t_\ell - t_\ell^A)R_\ell,
\end{aligned}
$$

while $\hat{\gamma}_{k,\ell,r}(|\mathcal{K}|) \geq \hat{\gamma}_{k,\ell,r}(\mathcal{K})$ if $k \in [E_d]$; $\hat{\gamma}_{k,\ell,r}(|\mathcal{K}|) \leq \hat{\gamma}_{k,\ell,r}(\mathcal{K})$ otherwise. It yields

$$\Delta P(\hat{\gamma}_{1,\ell,r}(\mathcal{K}), \ldots, \hat{\gamma}_{K,\ell,r}(\mathcal{K})) \leq \Delta P(\hat{\gamma}_{1,\ell,r}(|\mathcal{K}|), \ldots, \hat{\gamma}_{K,\ell,r}(|\mathcal{K}|)) \quad (44)$$

Note that each group generated by function GROUP has at most $\lceil|\mathcal{D}|/r\rceil + 1$ distinct elements, which corresponds to at most $\lceil|\mathcal{D}|/r\rceil + 1$ distinct elements by running function SINGLE-DEMAND, i.e., $|\mathcal{K}| \leq \lceil|\mathcal{D}|/r\rceil + 1$. We have then $\hat{\gamma}_{k,\ell,r}(\lceil|\mathcal{D}|/r\rceil + 1) \geq \hat{\gamma}_{k,\ell,r}(\mathcal{K})$, $\forall k \in [K]$. With (44),

$$\Delta P(\hat{\gamma}_{1,\ell,r}(\mathcal{K}), \ldots, \hat{\gamma}_{K,\ell,r}(\mathcal{K}))$$
$$\leq \Delta P(\hat{\gamma}_{1,\ell,r}(\lceil|\mathcal{D}|/r\rceil + 1), \ldots, \hat{\gamma}_{K,\ell,r}(\lceil|\mathcal{D}|/r\rceil + 1)).$$

Following the same procedure with all the groups, we can lower bound the total transmission power to satisfy demand combination $\mathbf{d}$ as follows

$$P(M, \boldsymbol{\pi}, \mathcal{D}) \leq \sum_{k=1}^{K} \left(\frac{2^{2\overline{\rho}_k} - 1}{h_k^2}\right) \prod_{j=1}^{k-1} 2^{2\overline{\rho}_j},$$

$$\overline{\rho}_k \triangleq \sum_{\ell=1}^{N} \sum_{r=\max\{\ell-N+|\mathcal{D}|,1\}}^{\min\{\ell,|\mathcal{D}|\}} \binom{N - |\mathcal{D}|}{\ell - r}\binom{|\mathcal{D}|}{r}$$
$$\cdot \hat{\gamma}_{k,\ell,r}(\lceil|\mathcal{D}|/r\rceil + 1),$$

which by letting $\mathcal{D} = [\min\{N, K\}]$, proves Theorem 1.

APPENDIX B
PROOF OF THEOREM 2

For a demand vector $\mathbf{d}$, $\forall \mathbf{d} \in [N]^K$, the proposed scheme presented in Section IV-C constructs a $N_e(\mathbf{d})$-level Gaussain superposition code. We denote the minimum total transmission power required by this scheme to satisfy $\mathbf{d}$ by $P(\mathbf{d}, M) = \sum_{i=1}^{N_e(\mathbf{d})} P_i(\mathbf{d}, M)$, where $P_i(\mathbf{d}, M)$ is the power allocated to generate the ith level codeword. With (24) and (25), we have

$$|V_{i,\mathbf{d}}^r| = M, \quad |V_{i,\mathbf{d}}^c| = \rho_i - M, \text{ if } k_i = i,$$
$$|V_{i,\mathbf{d}}^r| = 0, \quad |V_{i,\mathbf{d}}^c| = \rho_i, \text{ if } k_i \neq i.$$

Thus, according to (26) and (27), it yields

$$P_i(\mathbf{d}, M) = \begin{cases} \max\left\{\left(\dfrac{2^{2\tilde{\rho}_i} - 1}{h_i^2}\right)\left(1 + h_i^2 \sum_{j=i+1}^{K} P_j\right),\right. \\ \left.\left(\dfrac{2^{2(\tilde{\rho}_i+M)} - 1}{h_{i+1}^2}\right)\left(1 + h_{i+1}^2 \sum_{j=i+1}^{K} P_j\right)\right\}, \\ \quad \text{if } k_i = i, \\ \left(\dfrac{2^{2(\tilde{\rho}_i+M)} - 1}{h_i^2}\right)\left(1 + h_i^2 \sum_{j=i+1}^{K} P_j\right), \\ \quad \text{if } k_i \neq i, \end{cases}$$

$\forall i \in [N_e(\mathbf{d})]$. It is straightforward to see that the worst-case demand combination $\mathbf{d}_{\text{worst}}$ that maximizes $P(\mathbf{d}, M)$, i.e., $\mathbf{d}_{\text{worst}} = \arg\max_{\mathbf{d}} P(\mathbf{d}, M)$, is such that $N_e(\mathbf{d}) = \min\{N, K\}$ and $\mathcal{U} = [\min\{N, K\}]$, i.e., the weakest $\min\{N, K\}$ users request distinct files. And we have $P(\mathbf{d}_{\text{worst}}, M) = P_{UB}^{\text{PB}}(M)$, which completes the proof of Theorem 2.

REFERENCES

[1] Q. Yang, P. Hassanzadeh, D. Gündüz, and E. Erkip, "Centralized caching and delivery of correlated contents over a Gaussian broadcast channel," in *Proc. 16th Int. Symp. Modeling Optim. Mobile, Ad Hoc, Wireless Netw. (WiOpt)*, Shanghai, China, May 2018, pp. 1–6.

[2] S. O. Somuyiwa, A. György, and D. Gündüz, "A reinforcement-learning approach to proactive caching in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 6, pp. 1331–1344, Jun. 2018.

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

[4] M. M. Amiri, Q. Yang, and D. Gündüz, "Coded caching for a large number of users," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Cambridge, U.K., Sep. 2016, pp. 171–175.

[5] P. Hassanzadeh, A. Tulino, J. Llorca, and E. Erkip, "Correlation-aware distributed caching and coded delivery," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Cambridge, U.K., Sep. 2016, pp. 166–170.

[6] Q. Yang and D. Gündüz, "Centralized coded caching of correlated contents," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kansas City, MO, USA, May 2018, pp. 1–6.

[7] P. Hassanzadeh, A. M. Tulino, J. Llorca, and E. Erkip, "Broadcast caching networks with two receivers and multiple correlated sources," in *Proc. IEEE Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, USA, Oct./Nov. 2017, pp. 1449–1454.

[8] P. Hassanzadeh, A. M. Tulino, J. Llorca, and E. Erkip, "Rate-memory trade-off for the two-user broadcast caching network with correlated sources," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, Aachen, Germany, Jun. 2017, pp. 1227–1231.

[9] P. Hassanzadeh, A. M. Tulino, J. Llorca, and E. Erkip, "Cache-aided coded multicast for correlated sources," in *Proc. 9th Int. Symp. Turbo Codes Iterative Inf. Process. (ISTC)*, Brest, France, Sep. 2016, pp. 360–364.

[10] 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.

[11] M. Mohammadi Amiri and D. Gündüz, "Cache-aided content delivery over erasure broadcast channel," *IEEE Trans. Commun.*, vol. 66, no. 1, pp. 370–381, Jan. 2018.

[12] M. M. Amiri and D. Gündüz, "Caching and coded delivery over Gaussian broadcast channels for energy efficiency," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 8, pp. 1706–1720, Aug. 2018.

[13] S. S. Bidokhti, M. Wigger, and R. Timo, "Noisy broadcast networks with receiver caching," May 2016, *arXiv:1605.02317*. [Online]. Available: https://arxiv.org/abs/1605.02317

[14] S. S. Bidokhti, M. Wigger, and A. Yener, "Benefits of cache assignment on degraded broadcast channels," Feb. 2017, *arXiv:1702.08044*. [Online]. Available: https://arxiv.org/abs/1702.08044

[15] A. Ghorbel, M. Kobayashi, and S. Yang, "Content delivery in erasure broadcast channels with cache and feedback," *IEEE Trans. Inf. Theory*, vol. 62, no. 11, pp. 6407–6422, Nov. 2016.

[16] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. IT-19, no. 4, pp. 471–480, Jul. 1973.

[17] R. M. Gray and A. D. Wyner, "Source coding for a simple network," *Bell Syst. Tech. J.*, vol. 53, no. 9, pp. 1681–1721, 1974.

[18] P. Hassanzadeh, A. M. Tulino, J. Llorca, and E. Erkip, "Rate-memory trade-off for caching and delivery of correlated sources," Jun. 2018 *arXiv:1806.07333*. [Online]. Available: https://arxiv.org/abs/1806.07333

[19] T. S. Han, "The capacity region of general multiple-access channel with certain correlated sources," *Inf. Control*, vol. 40, no. 1, pp. 37–60, 1979.

[20] Z. Chen, P. Fan, and K. B. Letaief, "Fundamental limits of caching: Improved bounds for users with small buffers," *IET Commun.*, vol. 10, no. 17, pp. 2315–2318, Nov. 2016.

[21] Q. Yang and D. Gündüz, "Coded caching and content delivery with heterogeneous distortion requirements," *IEEE Trans. Inf. Theory*, vol. 64, no. 6, pp. 4347–4364, Jun. 2018.

[22] P. Bergmans, "A simple converse for broadcast channels with additive white Gaussian noise," *IEEE Trans. Inf. Theory*, vol. IT-20, no. 2, pp. 279–280, Mar. 1974.

[23] P. Bergmans, "Random coding theorem for broadcast channels with degraded components," *IEEE Trans. Inf. Theory*, vol. IT-19, no. 2, pp. 197–207, Mar. 1973.

[24] R. G. Gallager, "Capacity and coding for degraded broadcast channels," *Problemy Peredachi Informatsii*, vol. 10, no. 3, pp. 3–14, 1974.

Qianqian Yang (S'12–M'19) received the Ph.D. degree in electrical engineering from the Imperial College London in 2019. She is currently a Post-Doctoral Research Associate with the Imperial College London. Her research interests include communications and information theory, signal processing, and machine learning.

Parisa Hassanzadeh received the B.Sc. degree in electrical engineering from the Sharif University of Technology, Tehran, Iran, in 2013, and the Ph.D. degree in electrical engineering from New York University Tandon School of Engineering, Brooklyn, NY, USA, in 2019. She is currently a Research Scientist with the J. P. Morgan AI Research, New York, NY, USA. Her research interests include wireless communications and information theory, specifically on caching for wireless video delivery, and applications for the next generation of cellular systems.

Deniz Gündüz (S'03–M'08–SM'13) received the B.S. degree in electrical and electronics engineering from METU, Turkey, in 2002, and the M.S. and Ph.D. degrees in electrical engineering from the NYU Tandon School of Engineering (formerly Polytechnic University) in 2004 and 2007, respectively.

He served as a Post-Doctoral Research Associate with Princeton University, and a Consulting Assistant Professor with Stanford University. He was a Research Associate with CTTC, Barcelona, Spain, until September 2012, when he joined the Electrical and Electronic Engineering Department, Imperial College London, U.K., where he is currently a Reader (an Associate Professor) in information theory and communications, and leads the Information Processing and Communications Laboratory (IPC-Lab). His research interests include in the areas of communications and information theory, machine learning, and privacy.

He was a recipient of the IEEE Communications Society–Communication Theory Technical Committee (CTTC) Early Achievement Award in 2017, a Starting Grant of the European Research Council (ERC) in 2016, the IEEE Communications Society Best Young Researcher Award for the Europe, Middle East, and Africa Region in 2014, the Best Paper Award at the 2016 IEEE Wireless Communications and Networking Conference (WCNC), and the Best Student Paper Awards at the 2018 IEEE Wireless Communications and Networking Conference (WCNC), and the 2007 IEEE International Symposium on Information Theory (ISIT). He was the General Co-Chair of the 2012 European School of Information Theory, the 2016 IEEE Information Theory Workshop, the 2018 International ITG Workshop on Smart Antennas, and the 2019 London Symposium on Information Theory. He served as the Guest Editor of IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, Special Issue on Machine Learning in Wireless Communication, and an Editor of IEEE TRANSACTIONS ON COMMUNICATIONS from 2013 to 2018. He is an Editor of IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING. He is also a Distinguished Lecturer for the IEEE Information Theory Society.

Elza Erkip (S'93–M'96–SM'05–F'11) received the B.S. degree in electrical and electronics engineering from Middle East Technical University, Ankara, Turkey, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, USA.

She is currently an Institute Professor with the Electrical and Computer Engineering Department, New York University Tandon School of Engineering. Her research interests include information theory, communication theory, and wireless communications. She has been a member of the Board of Governors of the IEEE Information Theory Society since 2012, where she was the Society President in 2018. She is a member of the Science Academy of Turkey and among Clarivate Highly Cited Researchers. She was a recipient of the NSF CAREER Award in 2001, the IEEE Communications Society WICE Outstanding Achievement Award in 2016, and the IEEE Communications Society Communication Theory Technical Committee (CTTC) Technical Achievement Award in 2018. Her article awards include the IEEE Communications Society Stephen O. Rice Paper Prize in 2004, the IEEE Communications Society Award for Advances in Communication in 2013, and the IEEE Communications Society Best Tutorial Paper Award in 2019. She was a Distinguished Lecturer of the IEEE Information Theory Society from 2013 to 2014.