Optimizing Item and Subgroup Configurations for Social-Aware VR Shopping

Shao-Heng Ko*, Hsu-Chao Lai*†, Hong-Han Shuai†,
De-Nian Yang*, Wang-Chien Lee‡, Philip S. Yu§
*Academia Sinica, Taiwan †National Chiao Tung University, Taiwan
‡The Pennsylvania State University, USA §University of Illinois at Chicago, USA
*{arsenefrog, hclai0806, dnyang}@iis.sinica.edu.tw †hhshuai@g2.nctu.edu.tw
‡wlee@cse.psu.edu §psyu@cs.uic.edu

ABSTRACT

Shopping in VR malls has been regarded as a paradigm shift for E-commerce, but most of the conventional VR shopping platforms are designed for a single user. In this paper, we envisage a scenario of VR group shopping, which brings major advantages over conventional group shopping in brick-and-mortar stores and Web shopping: 1) configure flexible display of items and partitioning of subgroups to address individual interests in the group, and 2) support social interactions in the subgroups to boost sales. Accordingly, we formulate the Social-aware VR Group-Item Configuration (SVGIC) problem to configure a set of displayed items for flexibly partitioned subgroups of users in VR group shopping. We prove SVGIC is NP-hard and APX-hard. We design an approximation algorithm based on the idea of Co-display Subgroup Formation (CSF) to configure proper items for display to different subgroups of friends. Experimental results on real VR datasets and a user study with hTC VIVE manifest that our algorithms outperform baseline approaches by at least 30.1% in terms of solution quality.

PVLDB Reference Format:
Shao-Heng Ko, Hsu-Chao Lai, Hong-Han Shuai, De-Nian Yang, Wang-Chien Lee, Philip S. Yu. Optimizing Item and Subgroup Configurations for Social-Aware VR Shopping. *PVLDB*, 12(xxx): xxxx-yyyy, 2019.
DOI:

1. INTRODUCTION

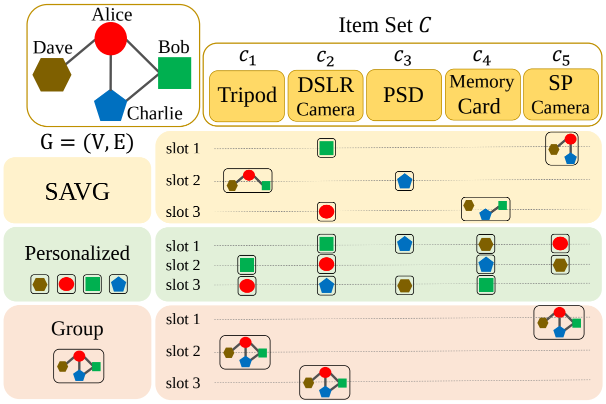
Virtual Reality (VR) has emerged as a disruptive technology for social [41], travel [18], and E-commerce applications. Particularly, shopping in VR malls is regarded as a paradigm shift for E-commerce stores, evident by emerging VR stores such as Amazon's VR kiosks [53], eBay and Myer's VR department store [17], Alibaba Buy+ [15], and IKEA VR Store [51]. Although these VR shopping platforms look promising, most of them are designed only for a single user instead of a group of friends, who also often appear in brick-and-mortar stores. As a result, existing approaches for configuring the displayed items in VR shopping malls are based on personal preference (similar to online web shopping) without considering potential social discussions amongst friends on items of interests, which is reported as beneficial for boosting sales in the marketing literature [56–58]. In this paper, with the support of *Customized Interactive Display* (CID), we envisage the scenario of group shopping with families and friends in the next-generation VR malls, where item placement is customized flexibly in accordance with both user preferences and potential social interactions during shopping.[1]

The CID technology [23, 32] naturally enables VR group shopping systems with two unique strengths: 1) *Customization.* Similar to group traveling and gaming in VR, the virtual environments (VEs) for individual users in VR group shopping need not be identical. While it is desirable to have consistent high-level layout and user interface for all users, the displayed items for each user can be customized based on her preferences. As CID allows different users to view different items at the same display slot, personalized recommendation is supported. 2) *Social Interaction.* While a specific slot no longer needs to display the same items to all users, users viewing a common item may engage a discussion on the item together, potentially driving up the engagement and purchase conversion [57, 58]. As a result, the displayed items could be tailored to maximize potential discussions during group shopping. In summary, compared with brick-and-mortar shopping, VR group shopping can better address the preferences of individuals in the group due to the new-found flexibility in item placement, which can be configured not only for the group as a whole but also for individuals and subgroups. On the other hand, compared with conventional E-commerce shopping on the Web, VR group shopping can boost sales by facilitating social interactions and providing an immersive experience.

In this paper, we tackle the challenging problem of configuring displayed items for *Social-Aware VR Group* (SAVG) shopping. Our strategy is to meet the item preferences of individual users via customization while enhancing potential discussions by displaying common items to a shopping group

[1]Customized display with different objects for each user has been widely adopted in VR shopping [51] and social [5] applications, e.g., in the IKEA demo [51] (at 0:40), the displayed furniture in the store may adapt to the preference of the user.

Proceedings of the VLDB Endowment, Vol. 12, No. xxx
ISSN 2150-8097.
DOI:



(a) Comparison of different approaches.

(b) Alice's view at slot 1.

(c) Alice's view at slot 2.

Figure 1: Illustrative example.

(and its subgroups). For customization, one possible approach is to use existing personalized recommendation techniques [13, 21, 22] to infer individual user preferences, and then retrieve the top-k favorite items for each user. However, this personalized approach fails to promote items of common interests that may trigger social interactions. To encourage social discussions, conventional group recommendation systems [24, 38, 44, 45] may be leveraged to retrieve a bundled itemset for all users. Nevertheless, by presenting the same configuration for the whole group of all users, this approach may sacrifice the diverse individual interests. In the following example, we illustrate the difference amongst the aforementioned approaches, in contrast to the desirable SAVG configuration we target on.

Example 1 (Illustrative Example)**.** Figure 1(a) depicts a scenario of group shopping for a VR store of digital photography. At the upper left is a social network $G = (V, E)$ of four VR users, Alice, Bob, Charlie, and Dave (indicated by red circles, green squares, blue pentagons, and brown hexagons, respectively). On top is an item set C consisting of five items: tripod, DSLR camera, portable storage device (PSD), memory card, and self-portrait (SP) camera. Given three display slots, the shaded areas, corresponding to different configuration approaches, illustrate how items are displayed to individuals or subgroups (in black rectangles), respectively. For instance, in SAVG, the SP camera is displayed at slot 1 to Alice, Charlie, and Dave to stimulate their discussion. Note that all slots for each user are filled for full utilization, e.g., for user Bob, the DSLR camera, tripod, and memory card are displayed at slots 1 to 3, respectively.

A configuration based on the personalized approach is shown in the light-green shaded area. It displays the top-3 items of interests to individual users based on their preferences (shown in Table 1 in Section 3). This configuration, aiming to increase the exposure of items of interests to customers, does not have users seeing the same item at the same slot. Next, the configuration shaded in light-orange, based on the group approach, displays exactly the same items to every user in the group. While encouraging discussions in the whole group, this configuration may sacrifice some individual interests and opportunities to sell some items, e.g., Dave may not find his favorite item (the memory card). Aiming to strike a balance between the factors of individual preferences and social interactions, the SAVG configuration forms subgroups flexibly across the displayed slots, having some users seeing the same items at the same slots (to encourage discussions), yet finding items of individual interests at the remaining slots. For example, the tripod is displayed at slot 2 to all users except for Charlie, who sees the PSD on his own. The DSLR camera is displayed to Bob at slot 1 and to Alice at slot 3, respectively, satisfying their individual interests. Figures 1(b) and 1(c) show Alice's view at slot 1 and slot 2, respectively, in this configuration. As shown, Alice is co-displayed the SP camera with Charlie and Dave at slot 1 (informed by the user icons below the primary view of the item), then co-displayed the Tripod with Bob and Dave at slot 2. As shown, SAVG shopping displays items of interests to individuals or different subgroups at each slot, and thereby is more flexible than other approaches.

□

As illustrated above, in addition to identifying which items to be displayed in which slots, properly partitioning subgroups by balancing both factors of personal preferences and social discussions is critical for the above-depicted SAVG shopping. In this work, we define the notion of *SAVG k-Configuration*, which *specifies the partitioned subgroups (or individuals) and corresponding items for each of the allocated k slots.* We also introduce the notion of *co-display* that represents users sharing views on common items. Formally, we formulate a new optimization problem, namely *<u>S</u>ocial-aware <u>V</u>R <u>G</u>roup-<u>I</u>tem <u>C</u>onfiguration* (SVGIC), to find the optimal SAVG k-Configuration that maximizes the overall 1) *preference utility* from users viewing their allocated items and 2) *social utility* from all pairs of friends having potential discussions on co-displayed items, where the basic preference utility of an individual user on a particular item and the basic social utility of two given users on a co-displayed item are provided as inputs. Meanwhile, we ensure that no duplicated items are displayed at different slots to a user (called *no-duplication* constraint). The problem is very challenging due to a complex trade-off between prioritizing personalized item configuration and facilitating social interactions. Notice that it is different from traditional community detection and group query problems, since 1) distinct displayed items (correlated to preference) and the discussion subgroup for each item (correlated to social tightness) need to be found simultaneously; 2) the displayed items and the discussion subgroups vary across all display slots; 3) the no-duplication constraint for all individual users is essential. Indeed, we prove that SVGIC is NP-hard and also APX-hard, i.e., it admits no polynomial-time approximation schemes (PTAS) unless $\mathsf{P} = \mathsf{NP}$.

To solve SVGIC, we first present an Integer Program (IP) as a baseline to find the exact solution which requires superpolynomial time. To address the efficiency issue while ensuring good solution quality, we then propose a novel approximation algorithm, namely *<u>A</u>lignment-aware <u>V</u>R Subgroup Formation* (AVG), to approach the optimal solution in polynomial time. Specifically, AVG first relaxes the IP to allow fragmented SAVG k-Configurations, and obtains the optimal (fractional) solution of the relaxed linear program. It then assigns the fractional decision variables derived from the optimal solution as the *utility factors* for various potential allocations of user-item-slot in the solution. Items of high utility factors are thus desirable as they are preferred to individuals or encouraging social discussions. Moreover, by leveraging an idea of *dependent rounding*, AVG introduces the notion of *Co-display Subgroup Formation* (CSF) to strike a balance between personal preferences and social interactions in forming subgroups. CSF forms a *target subgroup* of socially connected users with similar interests to display an item, according to a randomized *grouping threshold* on utility factors to determine the membership of the target subgroup. With CSF, AVG finds the subgroups (for all slots) and selects appropriate items simultaneously, and thereby is more effective than other approaches that complete these tasks separately in predetermined steps. Theoretically, we prove that AVG achieves 4-approximation in expectation. We then show that AVG can be derandomized into a deterministic 4-approximation algorithm for SVGIC.

Next, we enrich SVGIC by taking into account some practical VR operations and constraints. We define the notion of *indirect co-display* to capture the potential social utility obtained from friends displayed a common item at two different slots in their VEs, where discussion is facilitated via the *teleportation* [8] function in VR. We also consider a *subgroup size constraint* on the partitioned subgroups at each display slot due to practical limits in VR applications. Accordingly, we formulate the Social-aware VR Group-Item Configuration with Teleportation and Size constraint (SVGIC-ST) problem, and prove that SVGIC-ST is unlikely to be approximated within a constant ratio in polynomial time. Nevertheless, we extend AVG to support SVGIC-ST and guarantee feasibility of the solution.

The contributions of this work are as follows:

- We coin the notion of SAVG k-Configuration under the context of VR group shopping and formulate the SVGIC problem, aiming to return an SAVG k-Configuration that facilitates social interactions while not sacrificing the members' individual preferences. We prove SVGIC is APX-hard.
- We systematically tackle SVGIC by introducing an IP model and designing an approximation algorithm, AVG, based on the idea of Co-display Subgroup Formation (CSF) that leverages the flexibility of CID to partition subgroups for each slot and display common items to subgroup members.
- We define the SVGIC-ST problem to incorporate teleportation and subgroup size constraint. We prove SVGIC-ST admits no constant-ratio polynomial-time approximation algorithms unless ETH fails. We extend AVG to obtain feasible solutions of SVGIC-ST.
- A comprehensive evaluation on real VR datasets and a user study implemented in Unity and hTC VIVE manifest that our algorithms outperform the state-of-the-art recommendation schemes by at least 30.1% in terms of solution quality.

This paper is organized as follows. Section 2 reviews the related work. Section 3 formally defines the notion of SAVG k-Configuration, then formulates the SVGIC and SVGIC-ST problems and IP models for them. Then, Section 4 details the proposed AVG algorithm and the theoretical results. Section 5 reports the experimental results, and Section 6 concludes this paper.

2. RELATED WORK

Group Recommendation. Various schemes for estimating group preference have been proposed by aggregating features from different users in a group [38,44]. Cao *et al.* [10] propose an attention network for finding the group consensus. However, sacrificing personal preferences, the above group recommenders assign a unified set of items for the entire group based only on the aggregate preference without considering social topologies. For advanced approaches, recommendation-aware group formation [42] forms subgroups according to item preferences. SDSSel [45] finds dense subgroups and diverse itemsets. Shuai *et al.* [48] customize the sequence of shops without considering CID. However, the above recommenders find *static* subgroups, i.e., a universal partition, and still assign a fixed configuration of items to each subgroup, where every subgroup member sees the same item in the same slot. In contrast, the SAVG approach considered in this paper allows the partitioned subgroups to vary across all display slots and thereby is more flexible than the above works, whereas VR-specific operation, i.e., teleportation, can be supported in SAVG to facilitate discussions across different slots.

Personalized Recommendation. Personalized recommendation, a cornerstone of E-commerce, has been widely investigated. A recent line of study integrates deep learning with Collaborative Filtering (CF) [13,34] on heterogeneous applications [19], while Bayesian Personalized Ranking (BPR) [12,40] is proposed to learn the ranking of recommended items. However, the above works fail to consider social interactions in SAVG. While social relations have been leveraged to infer preferences for recommendations of products [60,61], POIs [28], and social events [31], they do not take into account the social interactions among users in recommendation of items. Thus, they fail to consider the tradeoff between social and personal factors under the context of social-aware group shopping. In this paper, we exploit the preferences obtained from such studies to serve as inputs for the tackled problems.

Social Group Search and Formation. Research on finding various groups from online social networks (OSNs) for different purposes has drawn increasing attention in recent years. Community search finds similar communities containing a given set of query nodes [14,29]. Group formation organizes a group of experts with low communication costs and specific skill combinations [7,39]. In addition, organizing a group in location-based social networks (LBSNs) based on certain spatial factors has also gained more attention [46,47]. Socio-Spatial Group Queries [46,55] find socially tight groups and select proper rally points for the groups. The problems tackled in the above studies are fundamentally different from the scenario in this paper, since they focus on retrieving only *parts* of the social network according to some criteria, whereas item selection across multiple slots is not addressed. Instead, SAVG group shopping aims to configure item display for all VR shopping users, whereas the subgroups partition the entire social network.

3. PROBLEM FORMULATION AND HARDNESS

In this section, we first define the notion of SAVG k-Configuration and then formally introduce the SVGIC and SVGIC-ST problems. We also prove their hardness of approximation, and introduce integer programs for SVGIC and SVGIC-ST as cornerstones for the AVG algorithm.

3.1 Problem Formulation

Given a collection $\mathcal{C}$ of m items (called the *Universal Item Set*), a directed social network $G = (V, E)$ with a vertex set V of n users (i.e., shoppers to visit a VR store) and an edge set E specifying their social relationships. In the following, we use the terms *user set* and *shopping group* interchangeably to refer to V, and define SAVG k-Configuration to represent the partitioned subgroups and the corresponding items displayed at their allocated slots.

Definition 1. *Social-Aware VR Group k-Configuration* (SAVG k-Configuration). Given k display slots for displaying items to users in a VR shopping group, an SAVG k-Configuration is a function $\mathbf{A}(\cdot,\cdot) : (V \times [k]) \to \mathcal{C}$ mapping a tuple (u, s) of a user u and a slot s to an item c. $\mathbf{A}(u, s) = c$ means that the configuration has item c displayed at slot s to user u, and $\mathbf{A}(u, :) = \langle \mathbf{A}(u, 1), \mathbf{A}(u, 2), ..., \mathbf{A}(u, k)\rangle$ are the k items displayed to u. Furthermore, the function is regulated by a *no-duplication constraint* which ensures the k items displayed to u are distinct, i.e., $\mathbf{A}(u, s) \neq \mathbf{A}(u, s'), \forall s \neq s'$.

For shopping with families and friends, previous research [56–58] demonstrates that discussions and interactions are inclined to trigger more purchases of an item. To display an item at the same slot to a pair of friends, we define the notion of *co-display* as follows.

Definition 2. *Co-display* ($u \overset{c}{\underset{s}{\leftrightarrow}} v$). Let $u \overset{c}{\underset{s}{\leftrightarrow}} v$ represent that users u and v are *co-displayed* an identical item c at slot s, i.e., $\mathbf{A}(u, s) = \mathbf{A}(v, s) = c$. Let $u \overset{c}{\leftrightarrow} v$ denote that there exists at least one s such that $u \overset{c}{\underset{s}{\leftrightarrow}} v$.

Naturally, the original group of users are partitioned into subgroups in correspondence with the displayed items. That is, for each slot $s \in [k]$, the SAVG k-Configuration implicitly partitions the user set V into a collection of $N_p(s)$ disjoint subsets $V^s = \{V_1^s, V_2^s, \cdots, V_{N_p(s)}^s\}$, such that $u \overset{c}{\underset{s}{\leftrightarrow}} v$ if and only if $u, v \in V_i^s, i = 1, \cdots, N_p(s)$.

Following previous research [33, 35, 43, 57], the satisfaction of a group shopping user consists of two factors: 1) her individual preference on the displayed items and 2) the satisfaction from sharing views and having potential discussions with friends. One intuitive way of modeling both factors is to maximize the aggregated preference while ensuring the aggregated social satisfaction is over a predefined threshold. However, this approach requires fine-tuning of the threshold.[2] In contrast, previous research [49, 54] has shown that a weighted combination of the preferential and social factors is effective in measuring user satisfaction. This formulation can also be viewed as a Lagrange Relaxation of the above constrained problem, and the best feasible solution of the constrained problem can thus be effectively approximated by scanning different combination weights.

Specifically, for a given pair of friends u and v, i.e., $(u, v) \in E$, and an item $c \in \mathcal{C}$, let $\mathrm{p}(u, c) \geq 0$ denote the *preference utility* of user u for item c, and let $\tau(u, v, c) \geq 0$ denote the *social utility* of user u from viewing item c together with user v, i.e., $u \overset{c}{\leftrightarrow} v$, where $\tau(u, v, c)$ can be different from $\tau(v, u, c)$.[3] Following the formulation in [49, 54], the *SAVG utility* is defined as follows.

Definition 3. *SAVG utility* ($\mathbf{w_A}(u, c)$). Given an SAVG k-Configuration $\mathbf{A}$, the SAVG utility of user u on item c in $\mathbf{A}$ represents a combination of the preference and social utilities, where $\lambda \in [0, 1]$ represents their relative weight.

$$\mathrm{w}_{\mathbf{A}}(u, c) = (1 - \lambda) \cdot \mathrm{p}(u, c) + \lambda \cdot \sum_{v | u \overset{c}{\leftrightarrow} v} \tau(u, v, c)$$

As mentioned, the preference and social utility values are given (e.g., learned from [31, 59]) as inputs. The weight λ can also be directly set by a user or implicitly learned from existing models [31, 59].[4] The impacts of different λ are studied later in Section 5.

Example 2. Revisit Example 1 where $\mathcal{C} = \{c_1, c_2, ..., c_5\}$ and let $V = \{u_A, u_B, u_C, u_D\}$ denote Alice, Bob, Charlie, and Dave, respectively. Table 1 summarizes the given preference and social utility values. For example, the preference utility of Alice to the tripod is $\mathrm{p}(u_A, c_1) = 0.8$. In the SAVG 3-configuration described at the top of Figure 1, $\mathbf{A}(u_A, \cdot) = \langle c_5, c_1, c_2\rangle$, meaning that the SP camera, the tripod, and the DSLR camera are displayed to Alice at slots 1, 2, and 3, respectively. Let $\lambda = 0.4$. At slot 2, Alice is *co-displayed* the tripod with Bob and Dave. Therefore, $\mathbf{w_A}(u_A, c_1) = 0.6 \cdot 0.8 + 0.4 \cdot (0.2 + 0.2) = 0.64$. □

We then formally introduce the SVGIC problem as follows.

Problem: Social-aware VR Group-Item Configuration.
Given: A social network $G = (V, E)$, a universal item set $\mathcal{C}$, preference utility $\mathrm{p}(u, c)$ for all $u \in V$ and $c \in \mathcal{C}$, social utility $\tau(u, v, c)$ for all $(u, v) \in E$ and $c \in \mathcal{C}$, the weight λ, and the number of slots k.
Find: An SAVG k-Configuration $\mathbf{A}^*$ to maximize the total SAVG utility

$$\sum_{u \in V} \sum_{c \in \mathbf{A}^*(u,\cdot)} \mathrm{w}_{\mathbf{A}^*}(u, c),$$

where $\mathrm{w}_{\mathbf{A}^*}(u, c)$ is the SAVG utility defined in Definition 3, and the no-duplication constraint is ensured.

---

[2] A low threshold of social satisfaction could be too easy to meet and thereby not optimized enough, whereas a high threshold is prone to render the problem infeasible.

[3] These utilities can be directly given by the users or obtained from social-aware personalized recommendation learning models [31, 59]. Those models, learning from purchase history, are able to infer (u, c) and (u, v, c) tuples to relieve users from filling those utilities manually.

[4] Another approach is to generate the user and item embeddings and design a neural network aggregator to derive the total *SAVG utility*. However, this approach requires a huge amount of data to learn the parameters and still needs to generate candidates as the model input. Later our VR user study results verify that the proposed SAVG utility is highly representative of the user satisfaction in SAVG shopping (Spearman correlation of 0.835).

Table 1: Preference and social utility values in Example 2.

	$\mathrm{p}(u_A,\cdot)$	$\mathrm{p}(u_B,\cdot)$	$\mathrm{p}(u_C,\cdot)$	$\mathrm{p}(u_D,\cdot)$	$\tau(u_A, u_B,\cdot)$	$\tau(u_A, u_C,\cdot)$	$\tau(u_A, u_D,\cdot)$	$\tau(u_B, u_A,\cdot)$	$\tau(u_B, u_C,\cdot)$	$\tau(u_C, u_A,\cdot)$	$\tau(u_C, u_B,\cdot)$	$\tau(u_D, u_A,\cdot)$
c_1	0.8	0.7	0	0.1	0.2	0	0.2	0.2	0	0	0.1	0.3
c_2	0.85	1.0	0.15	0	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
c_3	0.1	0.15	0.7	0.3	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.05
c_4	0.05	0.2	0.6	1.0	0	0	0.05	0.05	0.2	0.05	0.2	0
c_5	1.0	0.1	0.1	0.95	0.05	0.3	0.2	0.05	0	0.3	0.05	0.25

Table 2: Notations used in SVGIC.

Symbol	Description
n	number of users
m	number of items
k	number of display slots
$G=(V,E)$	social network
V	user set/shopping group
E	(social) edge set
$\mathcal{C}$	Universal Item Set
$\mathbf{A}(\cdot,\cdot)$	SAVG k-Configuration
$\mathbf{A}^*$	optimal SAVG k-Configuration
$\mathrm{p}(u,c)$	preference utility
$\tau(u,v,c)$	social utility
$\mathrm{w}_{\mathbf{A}}(u,c)$	SAVG utility
λ	weight of social utility
$\mathrm{p}'(u,c)$	scaled preference utility
$u \overset{c}{\underset{s}{\leftrightarrow}} v$	Co-display (at slot s)
$u \overset{c}{\leftrightarrow} v$	Co-display (at some slot)
$N_p(s)$	user subgroups
V^s	collection of subgroups at slot s
$x^c_{u,s}$	decision variable of u displayed c at slot s
x^c_u	decision variable of u displayed c
$y^c_{e,s}$	decision variable of u,v co-displayed c at slot s
y^c_e	decision variable of u,v co-displayed c

Table 3: Notations used in SVGIC-ST.

Symbol	Description
$\mathrm{w'}_{\mathbf{A}}(u,c)$	SAVG utility with indirect co-display
$u \xleftrightarrow[s,s']{c} v$	Indirect Co-display (at slots s,s')
$u \xleftrightarrow[\mathrm{ind}]{c} v$	Indirect Co-display (at some slots)
$\mathrm{d_{tel}}$	discount factor
M	subgroup size constraint
z^c_e	decision variable of u,v co-displayed c

Finally, we remark here that the above additive objective function in SVGIC can be viewed as a generalization of more limited variations of objectives. For instance, some objectives in personal/group recommendation systems without considering social discussions, e.g., the AV semantics in [42],[5] are special cases of SVGIC where $\lambda = 0$. All used notations in SVGIC are summarized in Table 2.

In SVGIC, we characterize the merit of social discussions by the social utilities to a pair of users when they see the same item at the *same slot*. Notice that a discussion on a common item displayed at different slots in the VEs of these two users may still happen, even if it is less likely. *Teleportation* [8], widely used in VR tourism and gaming applications, allows VR users to directly transport between different positions in the VE. Thus, for a pair of users Alice and Bob co-displayed an item at different slots, as long as they are aware of where the item is displayed, one (or both) of them can teleport to the respective display slot of the item to trigger a remote discussion. Thus, we further propose a generalized notion of *indirect co-display* to consider the above-described scenario.

Definition 4. *Indirect Co-display.* ($u \xleftrightarrow[s,s']{c} v$). Let $u \xleftrightarrow[s,s']{c} v$ denote that users u and v are *indirectly* co-displayed an item c at slots s and s' respectively in their VEs, i.e., $\mathbf{A}(u,s) = \mathbf{A}(v,s') = c$. We use $u \xleftrightarrow[\mathrm{ind}]{c} v$ to denote that there exist different slots $s \neq s'$ such that $u \xleftrightarrow[s,s']{c} v$.

Note that we use $u \overset{c}{\leftrightarrow} v$ to denote *direct co-display*, i.e., there exists at least one s such that $u \overset{c}{\underset{s}{\leftrightarrow}} v$, while using $u \xleftrightarrow[\mathrm{ind}]{c} v$ to denote *indirect co-display.* Also note that these two events are mutually exclusive due to the no-duplication constraint. Also note that $u \xleftrightarrow[s,s]{c} v$ is equivalent to direct co-display: $u \overset{c}{\underset{s}{\leftrightarrow}} v$.

As social discussions on indirectly co-displayed items are less immersive and require intentional user manipulation, we introduce a discount factor $\mathrm{d_{tel}} < 1$ on teleportation to downgrade the social utility obtained via indirect co-display. Therefore, the total SAVG utility incorporating indirect co-display is as follows.

Definition 5. *SAVG utility with indirect co-display* ($\mathrm{w'}_{\mathbf{A}}(u,c)$). Given an SAVG k-Configuration $\mathbf{A}$, the *SAVG utility with indirect co-display* of user u on item c in $\mathbf{A}$ is defined as

$$\mathrm{w'}_{\mathbf{A}}(u,c) = (1-\lambda)\cdot \mathrm{p}(u,c) + \lambda \cdot \Big(\sum_{v \mid u \overset{c}{\leftrightarrow} v} \tau(u,v,c) + \sum_{v \mid u \xleftrightarrow[\mathrm{ind}]{c} v} \mathrm{d_{tel}} \cdot \tau(u,v,c) \Big).$$

Finally, we note that existing VR applications often do not accommodate an unlimited number of users in one shared

[5] Other optimization criteria in existing group recommenders, e.g., Least-Misery (LM) [42] that considers fairness, can be considered. Later we show that the regret ratio (a metric for fairness) of our proposed methods still outperforms other baselines on real datasets.

VE, e.g., VRChat [5] allows at most 16 users, and IrisVR [4] allows up to 12 users within the same VE. Therefore, we further incorporate a *subgroup size constraint* M as an upper bound on the number of users directly co-displayed the same item at the same location. More specifically, for all slot $s \in [k]$, all partitioned subgroups $V_i^s \in V^s$ contain no more than M users; or equivalently, given a fixed slot s and a fixed item c, $\mathbf{A}^*(u,s) = c$ for at most M different users u. In the following, we introduce the SVGIC-ST problem that incorporates indirect co-display and the subgroup size constraint.

Problem: Social-aware VR Group-Item Configuration with Teleportation and Size constraint
(SVGIC-ST).
Given: A social network $G = (V, E)$, a universal item set $\mathcal{C}$, preference utility $\mathrm{p}(u,c)$ for all u and c, social utility $\tau(u,v,c)$ for all u,v, and c, the weight λ, the number of slots k, the teleportation discount $\mathrm{d}_{\mathrm{tel}}$, and the subgroup size upper bound M.
Find: An SAVG k-Configuration $\mathbf{A}^*$ to maximize the total SAVG utility (with indirect co-display)

$$\sum_{u \in V} \sum_{c \in \mathbf{A}^*(u,\cdot)} \mathrm{w'}_{\mathbf{A}^*}(u,c),$$

such that the subgroup size constraint is satisfied, and $\mathrm{w'}_{\mathbf{A}^*}(u,c)$ is defined as in Definition 5.

Notations in SVGIC-ST are summarized in Table 3.

3.2 Hardness Result and Integer Program

In the following, we prove that SVGIC is NP-hard and also APX-hard, i.e., it does not admit any polynomial-time approximation scheme (PTAS) unless P = NP. We then prove that SVGIC-ST does not admit any constant-factor polynomial-time approximation algorithms, unless the Exponential Time Hypothesis (ETH) fails.

Theorem 1. SVGIC is NP-hard and APX-hard.

Proof. We prove the theorem with a gap-preserving reduction from the maximum $\mathcal{K}_3$-packing problem (Max-K3P) [11]. Given a graph $\hat{G} = (\hat{V}, \hat{E})$, Max-K3P aims to find a subgraph $\hat{H} \subseteq \hat{G}$ with the largest number of edges such that $\hat{H}$ is a union of vertex-disjoint edges and triangles. Given a Max-K3P instance $\hat{G} = (\hat{V}, \hat{E})$, we construct an SVGIC instance as follows. Let $G = \hat{G}$. For each edge $e = (u,v) \in \hat{E}$, we construct an item c_e. Let social utility $\tau(u,v,c_e) = \tau(v,u,c_e) = 0.5$. For each triangle Δ consisting of vertices u, v, and w in $\hat{G}$, we construct an item c_Δ. Let social utility $\tau(u,v,c_\Delta) = \tau(v,u,c_\Delta) = \tau(u,w,c_\Delta) = \tau(w,u,c_\Delta) = \tau(v,w,c_\Delta) = \tau(w,v,c_\Delta) = 0.5$. Let all other social utility values be 0. Also, let preference utility $p(u,c) = 0$ for all users u and items c. Finally, let $k = \lambda = 1$.

We first prove the sufficient condition. Assume there exists a feasible $\hat{H} \subseteq \hat{G}$ with x edges. For each disjoint edge $e = (u,v) \in \hat{H}$, let $\mathbf{A}(u,1) = \mathbf{A}(v,1) = c_e$ in SVGIC. Similarly, for each disjoint triangle Δ consisting of vertices u, v, and w in $\hat{H}$, let $\mathbf{A}(u,1) = \mathbf{A}(v,1) = \mathbf{A}(w,1) = c_\Delta$. Each edge in $\hat{H}$ achieves an SAVG utility of 1. Therefore, this configuration achieves a total SAVG utility of x. We then prove the necessary condition. If the optimal objective in the SVGIC instance is x, we let $\hat{H}$ include all edges $e = (u,v)$ such that $\mathbf{A}(u,1) = \mathbf{A}(v,1) = c$ and $\tau(u,v,c) = 0.5$ (Note that c could be c_e or some c_Δ). It is straightforward to verify that any connected component in $\hat{H}$ is either an edge or a triangle. As each edge in $\hat{H}$ contributes 1 to the objective in SVGIC, there are exactly x edges in $\hat{H}$. Finally, since Max-K3P is APX-hard [11], SVGIC is also APX-hard (and thus also NP-hard). ☐

Based on SVGIC, we have the following observation on several special cases of this problem. The personalized approach, introduced in Section 1, is a special case of SVGIC with λ set to 0. In this special case, it is sufficient to leverage personalized (or social-aware personalized) recommendation systems to infer the personal preferences, because the problem reduces into trivially recommending the top-k favorite items for each user. On the other hand, the group approach is also a special case of SVGIC with $N_p(s) = 1$ for each s, i.e., all users view the same item at the same slot. However, the combinatorial complexity of the solution space for SVGIC is $\Theta(m^{nk})$ due to the flexible partitioning of subgroups across different display slots in SAVG. Formally, $\mathbf{A}(u,s) = \mathbf{A}(v,s), \forall u,v \in V, s = 1, \cdots, k$, which corresponds to a group recommendation scenario.

Due to the complicated interplay and trade-off of preference and social utility, the following theorem manifests that the optimal objective value of SVGIC can be significantly larger than the optimal objective values of the special cases above, corresponding to personalized and group approaches. Specifically, for each problem instance $\mathbf{I}$, let $\mathrm{OPT}(\mathbf{I})$ denote the optimal objective value of SVGIC. Let $\mathrm{OPT_P}(\mathbf{I})$ and $\mathrm{OPT_G}(\mathbf{I})$ respectively denote the optimal objective values of the special cases of personalized and group approaches in the above discussion, i.e., SVGIC with $\lambda = 1$, and SVGIC with $N_p(s) = 1$ for each slot s, respectively.

Theorem 2. Given any n, there exists 1) a SVGIC instance $\mathbf{I}_\mathrm{G}$ with n users such that $\frac{\mathrm{OPT}(\mathbf{I}_\mathrm{G})}{\mathrm{OPT_G}(\mathbf{I}_\mathrm{G})} = n$. 2) a SVGIC instance $\mathbf{I}_\mathrm{P}$ such that $\frac{\mathrm{OPT}(\mathbf{I}_\mathrm{P})}{\mathrm{OPT_P}(\mathbf{I}_\mathrm{P})} \geq \frac{\lambda}{1-\lambda} \cdot \frac{n-1}{2} = O(n)$.

Proof. We construct $\mathbf{I}_\mathrm{G}$ and $\mathbf{I}_\mathrm{P}$ as follows. For $\mathbf{I}_\mathrm{G}$, each user u_i in $\mathbf{I}_\mathrm{G}$ prefers exactly k items $\mathcal{C}_i = \{c_i, c_{n+i}, \cdots, c_{(k-1)n+i}\}$, i.e., $\mathrm{p}(u_i, c_j) = 1$ for $c_j \in \mathcal{C}_i$, and 0 for $c_j \notin \mathcal{C}_i$. Let $E = \emptyset$, and $\tau(u,v,c)$ thereby is 0 for all (u,v,c). In the optimal solution of the group approach, each slot s contributes at most $1-\lambda$ to the total SAVG utility because every item is preferred by exactly one user. By contrast, the optimal solution in SVGIC recommends u_i all the k items in $\mathcal{C}_i$. Each u_i then contributes $k(1-\lambda)$ to the total SAVG utility, and thus $\frac{\mathrm{OPT}(\mathbf{I}_G)}{\mathrm{OPT_G}(\mathbf{I}_G)} = \frac{nk(1-\lambda)}{k(1-\lambda)} = n$.

For $\mathbf{I}_\mathrm{P}$, each user u_i in $\mathbf{I}_\mathrm{P}$ prefers the items in $\mathcal{C}_i$ only *slightly more than* all the other items. Specifically, let $\mathrm{p}(u_i, c_j) = 1$ for $c_j \in \mathcal{C}_i$, and $\mathrm{p}(u_i, c_j) = 1 - \epsilon$ for $c_j \notin \mathcal{C}_i$. Let G represent a complete graph here, and $\tau(u,v,c) = 1$ for all u, v and c. The optimal solution of the personalized approach selects all the k items in $\mathcal{C}_i$ for each u_i and contributes $(1-\lambda) \cdot n \cdot k$ to the total SAVG utility. Note that co-display is not facilitated because no common item is chosen. By contrast, one possible solution in SVGIC recommends to all users the same k items in $\mathcal{C}_1$, where each item contributes $(1-\lambda) \cdot ((1-\epsilon)n + \epsilon)$ to the total preference and $\lambda \cdot \frac{n(n-1)}{2}$ to the social utility. Thus, as ϵ approaches 0,

$\frac{\mathrm{OPT}(\mathbf{I}_P)}{\mathrm{OPT}_{\mathrm{P}}(\mathbf{I}_P)} \geq \frac{k[(1-\lambda)\cdot((1-\epsilon)n+\epsilon)+\lambda\cdot\frac{n(n-1)}{2}]}{(1-\lambda)\cdot n\cdot k} \approx 1+\frac{\lambda}{1-\lambda}\cdot\frac{n-1}{2} = O(n)$, regarding λ as a constant. □

We then prove the hardness of approximation for the advanced SVGIC-ST problem.

Theorem 3. There exists no polynomial-time algorithm that approximates SVGIC-ST within any constant factor, unless the exponential time hypothesis (ETH) fails.

Proof. We prove the theorem with a gap-preserving reduction from the Densest k-Subgraph problem (DkS) [36]. Given a graph $\hat{G}=(\hat{V},\hat{E})$, DkS aims to find a subgraph $\hat{H}\subseteq\hat{G}$ with $\hat{k}$ vertices with the largest number of induced edges. (Note that for a fixed $\hat{k}$, maximizing the density is equivalent to maximizing the number of edges. We use $\hat{k}$ instead of the conventional k to avoid confusion with the number of slots in SVGIC-ST.) Given a DkS instance $\hat{G}=(\hat{V},\hat{E})$ and $\hat{k}$, we construct an SVGIC instance as follows. Let $V=\hat{V}\cup S$, where $S=\{w_1,w_2,\cdots,w_{|S|}\}$ consists of $|S|=\hat{k}-(|\hat{V}| \bmod \hat{k})$ additional vertices if $\hat{k}\nmid|\hat{V}|$, and $S=\emptyset$ if $\hat{k}\mid|\hat{V}|$. Let $E=\hat{E}$. Therefore, all vertices in S are singletons. Let $\mathcal{C}=\{c_1,c_2,\cdots,c_m\}$, where $m=\frac{|V|}{\hat{k}}$. Let preference utility $p(u,c)=0$ for all $u\in V$ and $c\in\mathcal{C}$. For each edge $e=(u,v)\in\hat{E}$, let $\tau(u,v,c_1)=\tau(v,u,c_1)=0.5$. Let $\tau(u,v,c)=0$ for all $(u,v)\notin\hat{E}$ or $c\neq c_1$. Finally, let $k=1$, $\lambda=1$, and $M=\hat{k}$.

We first prove the sufficient condition. Assume that there exists a subgraph $\hat{H}\subseteq\hat{G}$ with $\hat{k}$ vertices and x edges. We construct an SVGIC-ST solution with objective x by co-displaying c_1 to all vertices in $\hat{H}$ (and thereby obtaining social utility of 1 on each edge) and then randomly partitioning the vertices not in $\hat{H}$ to $m-1$ sets of cardinality $\hat{k}$. We then prove the necessary condition. Since there are $m=\frac{|V|}{\hat{k}}$ items, and the subgroup size constraint is $M=\hat{k}$, it follows that each item is co-displayed to exactly $M=\hat{k}$ users in a feasible solution. Assume the optimal objective in the SVGIC-ST instance is x, then the $M=\hat{k}$ users co-displayed item c_1 collectively form a induced subgraph of exactly x edges. Let $\hat{H}$ be the induced subgraph formed by these vertices in $\hat{G}$, then $\hat{H}$ has exactly x edges. Finally, according to [36], assuming the exponential time hypothesis (ETH) holds, there is no polynomial-time algorithm that approximates DkS to within a constant factor; in fact, a stronger inapproximability of $n^{(1/\log\log n)^c}$ holds. The result for SVGIC-ST directly follows. □

Here we further point out that both the hardness results of SVGIC and SVGIC-ST are from a simple case where $k=1$, i.e., the considered problems are already very hard when only one display slot is in consideration. Therefore, the hardness of SVGIC and SVGIC-ST with general values of k may be even harder.

Next, we propose an Integer Programming (IP) model for SVGIC and SVGIC-ST to serve as the cornerstone for the approximation algorithm proposed later in Section 4. We begin with the IP for SVGIC. Let binary variable $x^c_{u,s}$ denote whether user u is displayed item c at slot s, i.e., $x^c_{u,s}=1$ if and only if $\mathbf{A}(u,s)=c$. Let x^c_u indicate whether u is displayed c at any slot in the SAVG k-Configuration. Moreover, for each pair of friends $e=(u,v)\in E$, let binary variable $y^c_{e,s}$ denote whether u and v are co-displayed item c at slot s, i.e., $y^c_{e,s}=1$ if and only if $u \overset{c}{\underset{s}{\leftrightarrow}} v$. Similarly, variable $y^c_e=1$ if and only if $u \overset{c}{\leftrightarrow} v$. The objective of SVGIC is specified as follows.

$$\max \sum_{u\in V}\sum_{c\in\mathcal{C}}[(1-\lambda)\cdot \mathrm{p}(u,c)\cdot x^c_u+\lambda\cdot\sum_{e=(u,v)\in E}(\tau(u,v,c)\cdot y^c_e)]$$

subject to the following constraints,

$$\sum_{s=1}^{k} x^c_{u,s}\leq 1,\quad \forall u\in V, c\in\mathcal{C} \tag{1}$$

$$\sum_{c\in\mathcal{C}} x^c_{u,s}=1,\quad \forall u\in V, s\in[k] \tag{2}$$

$$x^c_u=\sum_{s=1}^{k} x^c_{u,s},\quad \forall u\in V, c\in\mathcal{C} \tag{3}$$

$$y^c_e=\sum_{s=1}^{k} y^c_{e,s},\quad \forall e=(u,v)\in E, c\in\mathcal{C} \tag{4}$$

$$y^c_{e,s}\leq x^c_{u,s},\quad \forall e=(u,v)\in E, s\in[k], c\in\mathcal{C} \tag{5}$$

$$y^c_{e,s}\leq x^c_{v,s},\quad \forall e=(u,v)\in E, s\in[k], c\in\mathcal{C} \tag{6}$$

$$x^c_{u,s}, x^c_u, y^c_{e,s}, y^c_e\in\{0,1\},\quad \forall u\in V, e\in E, s\in[k], c\in\mathcal{C}. \tag{7}$$

Constraint 1 states that each item c can only be displayed at most once to a user u (i.e., the no-duplication constraint). Constraint 2 guarantees that each user u is displayed exactly one item at each slot s. Constraint 3 ensures that $x^c_u=1$ (u is displayed c in the configuration) if and only if there is exactly one slot s with $x^c_{u,s}=1$. Similarly, constraint 4 ensures $y^c_e=1$ if and only if $y^c_{e,s}=1$ for exactly one s. Constraints 5 and 6 specify the co-display, where $y^c_{e,s}$ is allowed to be 1 only if c is displayed to both u and v at slot s, i.e., $x^c_{u,s}=x^c_{v,s}=1$. Finally, constraint 7 ensures all decision variables are binary.

Next, for each $e=(u,v)\in E$, let binary variable z^c_e denote whether u and v are co-displayed (both directed *or* indirected) item c. Note that $y^c_e=1$ implies $z^c_e=1$. To avoid repetitive calculation of the social utility, the coefficient before y^c_e in the objective is modified to be $(1-\mathrm{d_{tel}})$, such that a social utility of $\tau(u,v,c)$ is obtained when $y^c_e=z^c_e=1$. The objective of SVGIC-ST is thus

$$\max \sum_{u\in V}\sum_{c\in\mathcal{C}}[(1-\lambda)\cdot \mathrm{p}(u,c)\cdot x^c_u+\lambda\cdot\sum_{e=(u,v)\in E}\tau(u,v,c) \cdot\left((1-\mathrm{d_{tel}})\cdot y^c_e+\mathrm{d_{tel}}\cdot z^c_e\right)]$$

subject to constraints 1 to 6 above, as well as the following constraints:

$$z^c_e\leq x^c_u,\quad \forall e=(u,v)\in E, c\in\mathcal{C} \tag{8}$$

$$z^c_e\leq x^c_v,\quad \forall e=(u,v)\in E, c\in\mathcal{C} \tag{9}$$

$$x^c_{u,s}, x^c_u, y^c_{e,s}, y^c_e, z^c_e\in\{0,1\},\quad \forall u\in V, e\in E, s\in[k], c\in\mathcal{C}. \tag{10}$$

Constraints 8 and 9 specify the co-display, where z^c_e is allowed to be 1 only if c is displayed to u and v in some (not necessarily the same) slots, i.e., $x^c_u=x^c_v=1$. Therefore, $y^c_e=z^c_e=1$ for the case of direct co-display. Constraint 10 is the integrality constraint. The other constraints follow from the basic SVGIC problem. Note that in the optimal solution, whenever $y^c_e=1$ for some edge e, the corresponding z^c_e will also naturally equal 1, as it is always better to have a larger z^c_e. Therefore, no additional constraint needs to be introduced to enforce this relation.

Table 4: Notations used in AVG.

Symbol	Description
X^*	optimal fractional solution
${x^*}^c_{u,s}$	optimal decision variable
(c, s, α)	set of focal parameters
$c/\hat{c}$	focal item
$s/\hat{s}$	focal slot
$u/\hat{u}$	current user/eligible user
α	grouping threshold
U	target subgroup
OPT	optimal SAVG utility
$\mathcal{R}$	achieved utility
$\mathcal{R}_{\text{per}}$	achieved preference utility
$\mathcal{R}_{\text{soc}}$	achieved social utility
S_{cur}	available display units
$S_{\text{tar}}(c, s, {x^*}^c_{u,s})$	display units to be filled
$S_{\text{fut}}(c, s, {x^*}^c_{u,s})$	display units not to be filled
$\text{ALG}(S_{\text{tar}}(c, s, {x^*}^c_{u,s}))$	SAVG utility gained by CSF
$\text{OPT}_{\text{LP}}(S_{\text{fut}}(c, s, {x^*}^c_{u,s}))$	expected future SAVG utility
w^c_e	$\tau(\hat{u}, \hat{v}, c) + \tau(\hat{v}, \hat{u}, c)$
r	balancing ratio

4. ALGORITHM DESIGN

In this section, we introduce the Alignment-aware VR Subgroup Formation (AVG) algorithm to tackle SVGIC. As shown in Example 1, the personalized and group approaches do not solve SVGIC effectively, as the former misses on social utility from co-display while the latter fails to leverage the flexibility of CID to preserve personal preference. An alternative idea, called the *subgroup* approach, is to first pre-partition the shopping group (i.e., the whole user set) into some smaller social-aware subgroups (e.g., using traditional community detection techniques), and then determine the displayed items based on preferences of the subgroups. While this idea is effective for social event organization [54] where each user is assigned to exactly one social activity, it renders the partitioning of subgroups static across all display slots in SVGIC, i.e., a user is always co-displayed common items only with other users in the same subgroup. Therefore, this approach does not fully exploit the CID flexibility, leaving some room for better results.

Instead of using a universal partition of subgroups as in the aforementioned subgroup approach, we aim to devise a more sophisticated approach that allows varied co-display subgroups across the display slots to maximize the user experience. Accordingly, we leverage Linear Programming (LP) relaxation strategies that build on the solution of the Integer Program formulated in Section 3.2 because it naturally facilitates different subgroup partitions across all slots while allocating proper items for those subgroups with CID. In other words, our framework partitions the subgroups (for each slot) and selects the items simultaneously, thus avoiding any possible shortcomings of two-phased approaches that finish these two tasks sequentially. By relaxing all the integrality constraints in the IP, we obtain a relaxed linear program whose *fractional* optimal solution can be explicitly solved in polynomial time. For an item c to be displayed to a user u at a certain slot s, the fractional decision variable ${x^*}^c_{u,s}$ obtained from the optimal solution of the LP relaxation problem can be assigned as its *utility factor*. Items with larger utility factors are thus inclined to contribute more SAVG utility (i.e., the objective value), since they are preferred by the users or more capable of triggering social interactions.

Next, it is important to design an effective rounding procedure to construct a promising SAVG k-Configuration according to the utility factors. We observe that simple *independent* rounding schemes may perform egregiously in SVGIC because they do not facilitate effective co-displaying of common items, thereby losing a huge amount of potential social utility upon constructing the SAVG k-Configuration, especially in the cases where the item preferences are not diverse. Indeed, we prove that independent rounding schemes may achieve an expected total objective of only $O(1/m)$ of the optimal amount. Motivated by the incompetence of independent rounding, our idea is to leverage *dependent rounding* schemes that encourage co-display of items of common interests, i.e., with high utility factors to multiple users in the optimal LP solution.

Based on the idea of dependent rounding schemes in [26], we introduce the idea of *Co-display Subgroup Formation* (CSF) that co-displays a *focal item* c at a specific *focal slot* s to every user u with a utility factor $x^c_{u,s}$ greater than a *grouping threshold* α. In other words, CSF clusters the users with high utility factors to a focal item c to form a *target subgroup* in order to co-display c to the subgroup at a specific display slot s. Depending on the randomly chosen set of *focal parameters*, including the focal item, the focal slot, and the grouping threshold, the size of the created target subgroups can span a wide spectrum, i.e., as small as a single user and as large as the whole user set V, to effectively avoid the pitfalls of personalized and group approaches. The randomness naturally makes the algorithm less vulnerable to extreme-case inputs, therefore resulting in a good approximation guarantee. Moreover, CSF allows the partitions of subgroups to vary across all slots in the returned SAVG k-Configuration, exploiting the flexibility provided by CID. However, different from the dependent rounding schemes in [26], the construction of SAVG k-Configurations in SVGIC faces an additional challenge – it is necessary to carefully choose the displayed items at multiple slots to ensure the no-duplication constraint. Moreover, the sizes of partitioned VR subgroups need to be monitored in the advanced SVGIC-ST problem due to practical VR concerns (detailed in the full version [27]).

We prove that AVG is a 4-approximation algorithm for SVGIC and also show that AVG can be fully derandomized into a deterministic approximation algorithm. We also tailor CSF to consider the size constraint so as to extend AVG for the more complicated SVGIC-ST. In the following, we first deal with the case with $\lambda = \frac{1}{2}$. We observe that all other cases with $\lambda \neq 0$ can be reduced to this case by proper scaling of the inputs, i.e., $\mathrm{p}'(u, c) = \frac{1-\lambda}{\lambda}\mathrm{p}(u, c)$, whereas $\lambda = 0$ makes the problem become trivial. We explicitly prove this property in Section 4.4. Moreover, for brevity, the total SAVG utility is scaled up by 2 so that it is a direct sum of the preference and social utility. All used notations in AVG are summarized in Table 4.

4.1 LP Relaxation and an Independent Rounding Scheme

The LP relaxation problem of SVGIC can be formulated by replacing the integrality constraint (constraint 7) in the IP model with linear upper/lower bound constraints, i.e.,

Table 5: Optimal fractional solution for slot 1 in Example 2.

	${x^*}^{c_1}_{\cdot,1}$	${x^*}^{c_2}_{\cdot,1}$	${x^*}^{c_3}_{\cdot,1}$	${x^*}^{c_4}_{\cdot,1}$	${x^*}^{c_5}_{\cdot,1}$
u_A	0.33	0.33	0	0	0.33
u_B	0.33	0.33	0	0.33	0
u_C	0	0	0.33	0.33	0.33
u_D	0.33	0	0	0.33	0.33

$0 \leq x^c_{u,s}, x^c_u, y^c_{e,s}, y^c_e \leq 1$. The optimal fractional solution can be acquired in polynomial time with commercial solvers, e.g., CPLEX [3] or Gurobi [2]. Observe that for a set of fixed x variables, the largest objective is always achieved by setting every $y^c_{e,s}$ as large as possible, such that one of Constraints 5 and 6 in the relaxed LP becomes an equality (and thus $y^c_{e,s} = \min\{x^c_{u,s}, x^c_{v,s}\}$). Therefore, the optimal solution can be fully represented by X^* (the set of optimal x variables). The fractional decision variable ${x^*}^c_{u,s}$ in X^* is then taken as the *utility factor* of item c at slot s for user u. Note that the optimal objective in the relaxed LP is an upper bound of the optimal total SAVG utility in SVGIC, because the optimal solution in SVGIC is guaranteed to be a feasible solution of the LP relaxation problem.

Example 3. Table 5 shows the utility factors in Example 2, where the fractional solution is identical for all slots 1-3 (thereby only slot 1 is shown). For example, the utility factor of c_1 (the tripod) to Alice at each slot is ${x^*}^{c_1}_{u,1} = {x^*}^{c_1}_{u,2} = {x^*}^{c_1}_{u,3} = 0.33$.

Note that three items (c_1, c_2, and c_5) have nonzero utility factors to Alice at slot 1 in Example 3, which manifests that the optimal LP solution may not construct a valid SAVG k-Configuration because each user is allowed to display exactly one item at each display slot in SVGIC. Therefore, a rounding scheme is needed to construct appropriate SAVG k-Configurations from the utility factors. Given X^*, a simple rounding scheme is to randomly (and independently) assign item c to user u at slot s with probability ${x^*}^c_{u,s}$, i.e., the utility factor of c to u at s, so that more favorable items are more inclined to be actually displayed to the users. This rounding scheme is summarized in Algorithm 1.

Algorithm 1 Trivial Rounding Scheme

Input: X^*
Output: An SAVG k-Configuration **A**

for $u \in V$ **do**
  for $s \in \{1, 2, \cdots, k\}$ **do**
    Display item c to user u at slot s independently with probability ${x^*}^c_{u,s}$

However, as this strategy selects the displayed items independently, for a pair of friends u and v, the chance that the algorithm obtains high social utility by facilitating co-display is small, since it requires the randomized rounding process to hit on the same item for both u and v simultaneously. Furthermore, this strategy could not ensure the final SAVG k-Configuration to follow the no-duplication constraint, as an item c can be displayed to a user u at any slot with a nonzero utility factor. The following lemma demonstrates the ineffectiveness of this rounding scheme.

Lemma 1. There exists an SVGIC instance I on which the above rounding scheme achieves only a total SAVG utility of $O(\frac{1}{m})$ of the optimal value in expectation.

Proof. Assume that for all users $u, v \in V$ and $c \in \mathcal{C}$, we have $\mathrm{p}(u, c) = 0$ and $\tau(u, v, c) = \tau$ for a constant $\tau > 0$. Intuitively, every user is indifferent among all items. In this case, a trivial optimal solution for the relaxed LP can be found by setting ${x^*}^c_{u,s} = \frac{1}{m}$ for all c, u, s. As the trivial rounding scheme determines the displayed items independently, for any pair of users u, v and any slot s, the probability that u and v are co-displayed any item at slot s is only $\frac{1}{m}$. Therefore, the expected total SAVG utility achieved is $\frac{n(n-1)}{m} \cdot \tau \cdot k$. On the other hand, co-displaying an arbitrary item to all users achieves a social utility of $n(n-1) \cdot \tau$, and repeating this with distinct items for all slots yields a total SAVG utility of $n(n-1) \cdot \tau \cdot k$. This independent rounding scheme thus achieves only $\frac{(n-1)\tau}{m(n-1)\tau} = O(\frac{1}{m})$ of the optimal value. Moreover, the resulting SAVG k-Configuration is highly unlikely to satisfy the no-duplication constraint. □

4.2 Alignment-aware Algorithm

To address the above issues, our idea is to leverage the dependent rounding schemes for labeling problems in [26] to devise the *Co-display Subgroup Formation* (CSF) rounding scheme as the cornerstone of AVG, to find a target subgroup U according to a set of focal parameters for co-display of the focal item to all users in U. Given the optimal fractional solution X^* to the LP relaxation problem, AVG iteratively 1) samples a set of focal parameters (c, s, α) with $c \in \mathcal{C}, s \in \{1, 2, \cdots, k\}$, and $\alpha \in [0, 1]$ uniformly at random; it then 2) conducts CSF according to the selected set of parameters (c, s, α) until a complete SAVG k-Configuration is constructed. It is summarized in Algorithm 2.

Algorithm 2 Alignment-aware VR Subgroup Formation (AVG)

Input: X^*
Output: An SAVG k-Configuration **A**

$\mathbf{A}(\hat{u}, \hat{s}) \leftarrow$ NULL for all $\hat{u}, \hat{s}$
$X^* \leftarrow X^*_{\mathrm{LP}}$
while some entry in **A** is NULL **do**
  Sample $c \in \mathcal{C}$, $s \in [k]$, $\alpha \in [0, 1]$ randomly
  for $\hat{u} \in V$ **do**
    if $\mathbf{A}(\hat{u}, s) =$ NULL and $\mathbf{A}(\hat{u}, t) \neq c \, \forall t \neq s$ **then** ($\hat{u}$ eligible for (c, s))
      if ${x^*}^c_{\hat{u},s} \geq \alpha$ **then**
        $\mathbf{A}(\hat{u}, s) \leftarrow c$
return A

Co-display Subgroup Formation. Given the randomly sampled set of parameters (c, s, α), CSF finds the target subgroup as follows. With the focal item c and the focal slot s, a user $\hat{u}$ is *eligible* for (c, s) in CSF if and only if 1) $\hat{u}$ has not been displayed any item at slot s, and 2) c has not been displayed to $\hat{u}$ at any slot. Users not eligible for (c, s) are not displayed any item in CSF to ensure the no-duplication constraint. For each eligible user $\hat{u}$, CSF selects c for $\hat{u}$ at slot s (i.e., $\mathbf{A}(\hat{u}, s) \leftarrow c$) if and only if ${x^*}^c_{\hat{u},s}$ is no smaller than the grouping threshold α. In other words, given (c, s, α), CSF co-displays c to a target subgroup U that consists of every eligible user $\hat{u}$ with ${x^*}^c_{\hat{u},s} \geq \alpha$. Therefore, the grouping threshold α plays a key role to the performance bound in the formation of subgroups. Later we prove that with the above strategy, for any pair of users u, v and any item c, $\Pr(u \overset{c}{\leftrightarrow} v) \geq \frac{y^{*c}_e}{4}$; or equivalently, the expected social utility of u from viewing c with v obtained in the final SAVG k-Configuration is at least a constant factor within that in the optimal LP solution.

AVG repeats the process of parameter sampling and CSF until a *feasible* SAVG k-Configuration is fully constructed, i.e., each user is displayed exactly one item at each slot, and the no-duplication constraint is satisfied.

Revisiting independence vs. dependence. Recall the troublesome input instance in the proof of Lemma 1 where independent rounding performs badly. By exploiting the dependent rounding scheme in CSF, since $x^{*c}_{u,s} = \frac{1}{m}$ for all c, u, s, upon the first time a grouping threshold $\alpha \leq \frac{1}{m}$ is sampled, CSF co-displays the focal item c to *every user* in the shopping group, which is the optimal solution. On the other hand, independent rounding could not facilitate co-displaying an item to all users as effectively.

Example 4. For Example 2 with the utility factors shown in Table 5, assume that the set of focal parameters are sampled as $(c, s, \alpha) = (c_1, 3, 0.06)$. Since $x^{*c_1}_{u_A,3} = x^{*c_1}_{u_B,3} = x^{*c_1}_{u_D,3} = 0.33 > 0.06 > x^{*c_1}_{u_C,3} = 0$, CSF co-displays the tripod to the subgroup {Alice, Bob, Dave} at slot 3. Note that this solution is not the SAVG configuration in Figure 1. Next, for the second set of parameters $(c, s, \alpha) = (c_4, 2, 0.22)$, {Bob, Charlie, Dave} is formed and co-displayed the memory card at slot 2, since $x^{*c_4}_{u_B,2} = x^{*c_4}_{u_C,2} = x^{*c_4}_{u_D,2} = 0.33 > 0.22 > x^{*c_4}_{u_A,2} = 0$. In the third iteration, RFS selects $(c, s, \alpha) = (c_3, 1, 0.04)$. As only $x^{*c_3}_{u_C,1} = 0.33$ is nonzero among the utility factors for c_3 at slot 1, CSF assigns PSD to {Charlie} alone at slot 1. Next, in iteration 4, $(c, s, \alpha) = (c_5, 3, 0.2)$. At this moment, only Charlie has not been assigned an item at slot 3 since the others are co-displayed the tripod earlier. Because utility factor $x^{*c_3}_{u_C,5} = 0.33 > 0.2$, the SP camera is displayed to {Charlie} at slot 3. Iteration 5 generates $(c, s, \alpha) = (c_5, 1, 0.31)$. For the users without displayed items at slot 1, only Alice and Dave (but not Bob) have their utility factors of the SP camera larger than 0.31. Thus {Alice, Dave} are co-displayed the item at slot 1. The final two iterations with $x^{*c_3}_{u_C,5}$ as $(c_2, 1, 0.01)$ and $(c_2, 2, 0.19)$ displays the DSLR camera to {Bob} at slot 1 and {Alice} at slot 2. This finalizes the construction of a SAVG k-Configuration as represented in Table 6, achieving a total SAVG utility of 9.75.[6] □

We then show that AVG is a 4-approximation algorithm for SVGIC in expectation.

Theorem 4. Given the optimal fractional solution X^*, AVG returns an expected 4-approximate SAVG k-Configuration in $O(n^2 \cdot k)$-time.

Let OPT be the optimal total SAVG utility in SVGIC. An *iteration* of AVG includes 1) focus phase sampling a (c, s, α) and 2) CSF with (c, s, α). Let $\mathcal{R}$ denote the total SAVG utility achieved by AVG. Moreover, let $\mathcal{R}_{\text{pre}}$ and $\mathcal{R}_{\text{soc}}$ denote the total preference and social utilities achieved by AVG, respectively. Let OPT be the optimal total SAVG utility of SVGIC. Let X and Y be the solutions found by AVG, i.e., feasible solutions to $x^c_{u,s}$, x^c_u, $y^c_{e,s}$, and y^c_e in Section 3.2. Let $\mathrm{w}^c_e = \tau(u, v, c) + \tau(v, u, c)$ for all $e = (u, v) \in E$. Therefore, $\mathbb{E}[\mathcal{R}] = \mathbb{E}[\mathcal{R}_{\text{pre}}] + \mathbb{E}[\mathcal{R}_{\text{soc}}]$, where

$$\mathbb{E}[\mathcal{R}_{\text{pre}}] = \mathbb{E}[\sum_{c \in \mathcal{C}} \sum_{u \in V} \mathrm{p}(u, c) \cdot x^c_u]$$

$$\mathbb{E}[\mathcal{R}_{\text{soc}}] = \mathbb{E}[\sum_{c \in \mathcal{C}} \sum_{e \in E} \mathrm{w}^c_e \cdot y^c_e]$$

We first prove that $\mathbb{E}[\mathcal{R}_{\text{per}}] \geq \sum_{c \in \mathcal{C}} \sum_{u \in V} \left(\mathrm{p}(u, c) \cdot \frac{x^{*c}_u}{2}\right)$. Based on the definition of AVG, we have the following observation.

Lemma 2. In any iteration t, if u is eligible for (c, s), the probability P^u_{rec} that $\mathbf{A}(u, s) \leftarrow c$ is $\frac{x^{*c}_{u,s}}{k \cdot m}$ (where k and m are respectively the numbers of slots and items) since c and s are selected randomly,[7] and α is uniformly chosen from $[0, 1]$.

Note that the above observation only gives the conditional probability of u being assigned c at slot s when u is still eligible for (c, s) at the beginning of iteration t. Thus, we also need to derive the probability that u is eligible for (c, s) for each iteration t.

Lemma 3. In any iteration t, for any c, s and a user u eligible for (c, s), the probability P^u_{ne} that u is not eligible for (c, s) in iteration $(t + 1)$ is at most $\frac{2}{k \cdot m}$.

Proof. User u is not eligible for (c, s) in iteration $(t + 1)$ when one of the following cases occurs in iteration t: 1) u is displayed c in some slot $\hat{s}$, or 2) u is displayed some item $\hat{c}$ at slot s. From Lemma 2, the probabilities for the above two cases are at most $\sum_{\hat{s}=1}^{k} \frac{x^{*c}_{u,\hat{s}}}{k \cdot m}$ and $\sum_{\hat{c} \in \mathcal{C}} \frac{x^{*\hat{c}}_{u,s}}{k \cdot m}$, respectively. Recall that for any u, $\sum_{\hat{s}=1}^{k} x^{*c}_{u,\hat{s}} \leq 1$ and $\sum_{\hat{c} \in \mathcal{C}} x^{*\hat{c}}_{u,s} = 1$ in LP relaxation. Therefore, the total probability of the above cases is at most $\frac{1}{k \cdot m}(\sum_{\hat{s}=1}^{k} x^{*c}_{u,\hat{s}} + \sum_{\hat{c} \in \mathcal{C}} x^{*\hat{c}}_{u,s}) \leq \frac{2}{k \cdot m}$. □

In the following, we first consider the case that u is eligible for (c, s) in the beginning of iteration t. According to Lemma 2, the probability P_{rec} that u is displayed c in slot s in this iteration is $P_{\text{rec}} = \frac{x^{*c}_{u,s}}{k \cdot m}$. Moreover, according to above, let P_{ne} denote the probability of u losing eligibility for (c, s) in this iteration, then $P_{\text{ne}} \leq \frac{2}{k \cdot m}$. Therefore, we have

$$\begin{aligned}
&\Pr(\mathbf{A}(u, s) = c) \\
&= \sum_{t=1}^{\infty} \Pr(\mathbf{A}(u, s) \leftarrow c \text{ in iteration } t) \\
&= \sum_{t=1}^{\infty} P^u_{\text{rec}} \cdot \Pr[u \text{ is eligible for } (c, s) \text{ in the } t\text{-th iteration}] \\
&= \sum_{t=1}^{\infty} P^u_{\text{rec}} \cdot (1 - P^u_{\text{ne}})^{t-1} = \frac{P^u_{\text{rec}}}{P^u_{\text{ne}}} \geq \frac{x^{*c}_{u,s}}{2},
\end{aligned}$$

[6] It is worth noting that those iterations with focal parameters not leading to any item display are omitted from this example. Indeed, it suffices for RFS to sample from the combinations of focal parameters that does result in item displays, which also help improve the practical efficiency of AVG. We revisit this issue in the proof of appoximation guarantee.

[7] Sampling a previously chosen (c, s) is allowed here because a smaller α may display c to more users.

where $t = \infty$ is allowed in the analysis (but not in the algorithm design) since an empty target group can be randomly generated here (explained later). Thus, $\mathbb{E}[\mathcal{R}_{\text{per}}] \geq \sum_{c\in\mathcal{C}} \sum_{u\in V} \left(\text{p}(u,c) \cdot \frac{x^{*c}_{u}}{2}\right)$. Next, we aim to use a similar approach to prove that $\mathbb{E}[\mathcal{R}_{\text{soc}}] \geq \frac{1}{4} \sum\limits_{c\in\mathcal{C}} \sum\limits_{e\in E} \text{w}^c_e \cdot y^{*c}_{e}$. To prove this for the more complicated social utility, instead of directly analyzing $\mathbb{E}[\mathcal{R}_{\text{soc}}]$, we first consider the case that the social utility $\tau(u,v,c)$ is generated when both u and v are co-displayed c in the same iteration, and let $\mathbb{E}[\mathcal{R}'_{\text{soc}}]$ denote the expected total social utility in this case. Clearly, $\mathbb{E}[\mathcal{R}_{\text{soc}}] \geq \mathbb{E}[\mathcal{R}'_{\text{soc}}]$. Similarly, we have the following observations.

Lemma 4. In any iteration t, for any pair of users $e = (u,v)$ with both u and v eligible for (c,s), the probability that $\mathbf{A}(u,s) \leftarrow c$ *or* $\mathbf{A}(v,s) \leftarrow c$ is $\frac{\max\{x^{*c}_{u,s}, x^{*c}_{v,s}\}}{k\cdot m}$, and the probability P^e_{rec} that $\mathbf{A}(u,s) \leftarrow c$ *and* $\mathbf{A}(v,s) \leftarrow c$ is $\frac{\min\{x^{*c}_{u,s}, x^{*c}_{v,s}\}}{k\cdot m}$.

The reason of Observation 4 is as follows. If $\alpha \leq \min\limits_{u\in U} x^{*c}_{u,s}$, CSF assigns c at s to all users in U since $x^c_{u,s} \geq \alpha$ for all $u \in U$. Similarly, if $\alpha \leq \max\limits_{u\in U} x^{*c}_{u,s}$, CSF at least assigns c to the user u with the largest $x^c_{u,s}$. For each iteration t, the following lemma then bounds the probability that at least one user in a group U *loses eligibility* for (c,s) in iteration t, either due to the no-duplication constraint or due to the assignment of some other item at slot s.

Lemma 5. In any iteration t, for any pair of users $e = (u,v)$ with u and v eligible for (c,s), the probability P^e_{ne} that *at least one* of u and v is not eligible for (c,s) in iteration $(t+1)$ is at most $\frac{4}{k\cdot m}$.

Proof. At least one of u and v is not eligible for (c,s) in iteration $(t+1)$ when one of the following cases occurs in iteration t: 1) u or v is displayed c in some slot $\hat{s}$, or 2) u or v is displayed some item $\hat{c}$ in slot s. From Lemma 4, the probabilities for the above two cases are at most $\sum\limits_{\hat{s}=1}^{k} \frac{\max\{x^{*c}_{u,\hat{s}}, x^{*c}_{v,\hat{s}}\}}{k\cdot m}$ and $\sum\limits_{\hat{c}\in\mathcal{C}} \frac{\max\{x^{*\hat{c}}_{u,s}, x^{*\hat{c}}_{v,s}\}}{k\cdot m}$, respectively. Recall that for any u, $\sum\limits_{\hat{s}=1}^{k} x^{*c}_{u,\hat{s}} \leq 1$ and $\sum\limits_{\hat{c}\in\mathcal{C}} x^{*\hat{c}}_{u,s} = 1$ in LP relaxation. Therefore, the total probability of the above cases is at most

$$\begin{aligned}
&\frac{1}{k\cdot m}\Big(\sum_{\hat{s}=1}^{k} \max\{x^{*c}_{u,\hat{s}}, x^{*c}_{v,\hat{s}}\} + \sum_{\hat{c}\in\mathcal{C}} \max\{x^{*\hat{c}}_{u,s}, x^{*\hat{c}}_{v,s}\}\Big)\\
&\leq \frac{1}{k\cdot m}\Big(\sum_{\hat{s}=1}^{k} (x^{*c}_{u,\hat{s}} + x^{*c}_{v,\hat{s}}) + \sum_{\hat{c}\in\mathcal{C}} (x^{*\hat{c}}_{u,s} + x^{*\hat{c}}_{v,s})\Big)\\
&= \frac{1}{k\cdot m}\Big(\sum_{\hat{s}=1}^{k} x^{*c}_{u,\hat{s}} + \sum_{\hat{s}=1}^{k} x^{*c}_{v,\hat{s}} + \sum_{\hat{c}\in\mathcal{C}} x^{*\hat{c}}_{u,s} + \sum_{\hat{c}\in\mathcal{C}} x^{*\hat{c}}_{v,s}\Big)\\
&\leq \frac{4}{k\cdot m}.
\end{aligned}$$

□

Similarly, consider the case that in the beginning of iteration t, both u,v are eligible for (c,s). Let $u \overset{c}{\underset{s}{\leftrightarrow}} v|_t$ denote that u,v are co-displayed c at slot s in iteration t. Therefore, we have

$$\begin{aligned}
&\Pr[u \overset{c}{\underset{s}{\leftrightarrow}} v] = \sum_{t=1}^{\infty} \Pr[u \overset{c}{\underset{s}{\leftrightarrow}} v|_t]\\
&= \sum_{t=1}^{\infty} P^e_{\text{rec}} \cdot \Pr[u,v \text{ are both eligible for } (c,s) \text{ in } t\text{-th iteration}]\\
&= \sum_{t=1}^{\infty} P^e_{\text{rec}} \cdot \Pr[u,v \text{ both eligible for } (c,s) \text{ in } t\text{-th iteration}]\\
&= \sum_{t=1}^{\infty} P^e_{\text{rec}} \cdot (1-P^e_{\text{ne}})^{t-1} = \frac{P^e_{\text{rec}}}{P^e_{\text{ne}}}\\
&\geq \frac{\frac{\min\{x^{*c}_{u,s}, x^{*c}_{v,s}\}}{k\cdot m}}{\frac{4}{k\cdot m}}\\
&\geq \frac{\min\{x^{*c}_{u,s}, x^{*c}_{v,s}\}}{4} = \frac{y^{*c}_{e,s}}{4}.
\end{aligned}$$

Finally, because of the no-duplicate constraint, for all u, v and c, the events $u \overset{c}{\underset{s}{\leftrightarrow}} v|_t$ and $u \overset{c}{\underset{s'}{\leftrightarrow}} v|_t$ for different slots s and s' are mutually exclusive. Similarly, the events $u \overset{c}{\underset{s}{\leftrightarrow}} v|_t$ and $u \overset{c}{\underset{s}{\leftrightarrow}} v|_{t'}$ for $t \neq t'$ are also mutually exclusive because every user sees exactly one item at each slot. Therefore,

$$\begin{aligned}
\Pr(u \overset{c}{\leftrightarrow} v) &\geq \sum_{s=1}^{k} \sum_{t=1}^{\infty} \Pr(u \overset{c}{\underset{s}{\leftrightarrow}} v|_t),\\
\mathbb{E}[\mathcal{R}_{\text{soc}}] &\geq \mathbb{E}[\mathcal{R}'_{\text{soc}}]\\
&= \sum_{c\in\mathcal{C}} \sum_{e\in E} \text{w}^c_e \cdot \sum_{t=1}^{\infty} \Pr(u \overset{c}{\underset{s}{\leftrightarrow}} v|_t)\\
&\geq \sum_{c\in\mathcal{C}} \sum_{e\in E} \text{w}^c_e \cdot \frac{y^{*c}_{e}}{4}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\mathbb{E}[\mathcal{R}] &= \mathbb{E}[\mathcal{R}_{\text{per}}] + \mathbb{E}[\mathcal{R}_{\text{soc}}]\\
&\geq \sum_{c\in\mathcal{C}} \sum_{u\in V} \text{p}(u,c) \cdot \frac{x^{*c}_{u}}{2} + \sum_{c\in\mathcal{C}} \sum_{e\in E} \text{w}^c_e \cdot \frac{y^{*c}_{e}}{4}\\
&\geq \frac{\text{OPT}}{4},
\end{aligned}$$

which proves the approximation ratio.

In the above derivation, a large α could lead to an empty target group if it exceeds $x^{*c}_{u,s}$ for every u eligible for (c,s). Therefore, the total number of iterations could approach ∞. To address the above issue, instead of sampling (c,s,α) uniformly from all possible combinations, AVG samples (c,s,α) uniformly from only the combinations generating nonempty target groups (i.e., an enormous α is no longer chosen). Because the setting of α is independent for each iteration, and the probability of selecting each combination to generate a nonempty target group remains equal, the expected solution quality is also identical for AVG. Therefore, the number of iterations for AVG is effectively reduced to $O(nk)$, and CSF in each iteration requires $O(n)$-time. The total time complexity of AVG, including the config phase, is thus

$O(\text{LP})+O(n^2 \cdot k)$, where $O(\text{LP})$ is the complexity[8] of solving X^*. $\square$ Two immediate corollaries are directly obtained from Theorem 4 as follows.

Corollary 4.1. Repeating AVG and selecting the best output returns a $(4+\epsilon)$-approximate SAVG k-Configuration in $O(n^2 \cdot k \cdot \log_\epsilon n)$-time with high probability, i.e., with a probability $1 - \frac{1}{n^{O(1)}}$.

Corollary 4.2. Given a (non-optimal) fractional solution $\tilde{X}^*$ as a β-approximation of the LP relaxation problem, AVG returns an expected $(4 \cdot \beta)$-approximate SAVG k-Configuration.

Proof. For the first corollary, from Theorem 4, AVG achieves an expected total SAVG utility $\mathbb{E}[\mathcal{R}] \geq \frac{OPT}{4}$. Let $\mathcal{R}' = OPT - \mathcal{R}$ denote the gap between $\mathcal{R}$ (the objective achieved by AVG) and OPT (the optimal objective). Clearly, $\mathcal{R}'$ is non-negative, and we have $\mathbb{E}[\mathcal{R}'] = \mathbb{E}[OPT - \mathcal{R}] = OPT - \mathbb{E}[\mathcal{R}] \leq \frac{3 \cdot OPT}{4}$. Therefore, by the Markov inequality, the probability that a single invocation of AVG failing to return a $(4+\epsilon)$-approximate SAVG k-Configuration is $\mathcal{R}' > (1 - \frac{1}{4+\epsilon}) \cdot OPT$ is

$$\begin{aligned}\Pr[\mathcal{R}' \geq OPT \cdot (1 - \frac{1}{4+\epsilon})] &\leq \frac{\mathbb{E}[\mathcal{R}']}{(1-\frac{1}{4+\epsilon}) \cdot OPT} \\ &\leq \frac{\frac{3 \cdot OPT}{4}}{(1-\frac{1}{4+\epsilon}) \cdot OPT} \\ &= \frac{12+3\epsilon}{12+4\epsilon}.\end{aligned}$$

Therefore, if AVG is repeated n_{repeat} times, the probability that the best solution is not a $(4+\epsilon)$-approximate SAVG k-Configuration is at most $(\frac{12+3\epsilon}{12+4\epsilon})^{n_{\text{repeat}}}$. By setting $n_{\text{repeat}} = \log_{\frac{12+4\epsilon}{12+3\epsilon}} n$, the success rate will be at least $1 - (\frac{12+3\epsilon}{12+4\epsilon})^{n_{\text{repeat}}} = 1 - \frac{1}{n}$.

A single run of AVG is in $O(n^2 \cdot k)$-time. Therefore, repeating it for $\log_{\frac{12+4\epsilon}{12+3\epsilon}} n$ times can be done in $O(n^2 \cdot k \cdot \log_{\frac{12+4\epsilon}{12+3\epsilon}} n) = O(n^2 \cdot k \cdot \log_\epsilon n)$-time.

Finally, for the second corollary, note that the total objective achieved by the non-optimal approximate LP solution $\tilde{X}^*$ is at least $\beta \cdot OPT$, since it is a β-approximation of the LP relaxation problem, which has an optimal objective of at least OPT. Therefore, from Theorem 4, running AVG with $\tilde{X}^*$ achieves an expected SAVG utility of at least $\frac{OPT}{4 \cdot \beta}$. $\square$

The second corollary is particularly useful in practice for improving the efficiency of AVG since state-of-the-art LP solvers often reach a close-to-optimal solution in a short time but need a relatively long time to return the optimal solution, especially for large inputs. Therefore, it allows for a quality-efficiency trade-off in solving SVGIC.

4.3 Derandomizing AVG

From the investigation of AVG, we observe that the grouping threshold α plays a key role in forming effective target subgroups in CSF. If α is close to 0, CSF easily forms a large subgroup consisting of all users and co-displays the focal item to them, similar to the ineffective group approach. On the other hand, large α values lead to small subgroups, not good for exploiting social interactions. Due to the randomness involved in AVG, these caveats cannot be completely avoided. To address these issues, we aim to further strengthen the performance guarantee of AVG by derandomizing the selection of focal parameters to obtain a stronger version of the algorithm, namely Deterministic Alignment-aware VR Subgroup Formation (AVG-D), which is a *deterministic* 4-approximation algorithm. First, we observe that α can be assigned in a discrete manner.

Observation 1. There are $O(knm)$ distinct possible outcomes in CSF, each corresponding to a grouping threshold $\alpha = {x^*}^c_{u,s}$, i.e., the utility factor of an item c to a user u at a slot s.

The above observation can be verified as follows. Given c and s, the outcome of CSF with grouping threshold $\alpha = x$, for any $x \in [0,1]$, is equivalent to that with α set to the smallest ${x^*}^c_{u,s} \geq x$. It enables us to derandomize AVG effectively. Instead of randomly sampling (c, s, α), we carefully evaluate the outcomes (of CSF) from setting α to every possible ${x^*}^c_{u,s}$ in the optimal fractional solution. Intuitively, it is desirable to select an α that results in the largest increment in the total SAVG utility. However, this short-sighted approach ignores the potentially significant increase in total SAVG utility in the future from the remaining users and slots that have not been processed. In fact, it always selects an outcome with $\alpha = 0$ to maximize the current utility increment. Therefore, it is necessary for AVG-D to carefully evaluate all potential future allocations of items.

Specifically, let $S_{\text{cur}} = \{(\hat{u}, \hat{s}) | \mathbf{A}(\hat{u}, \hat{s}) = \text{NULL}\}$ denote the set of *display units*, i.e., a slot $\hat{s}$ for a user $\hat{u}$, that have not been assigned an item before the current iteration. Let $S_{\text{tar}}(c, s, {x^*}^c_{u,s}) = \{(\hat{u}, s) \in S_{\text{cur}} | {x^*}^c_{\hat{u},s} \geq {x^*}^c_{u,s}\}$ denote the set of display units to be assigned c in CSF with $\alpha = {x^*}^c_{u,s}$. Let $S_{\text{fut}}(c, s, {x^*}^c_{u,s}) = S_{\text{cur}} \setminus S_{\text{tar}}(c, s, {x^*}^c_{u,s})$ denote the set of remaining display units to be processed in the future. Moreover, let $\text{ALG}(S_{\text{tar}}(c, s, {x^*}^c_{u,s}))$ be the SAVG utility gained by co-displaying c to the target subgroup. Let $\text{OPT}_{\text{LP}}(S_{\text{fut}}(c, s, {x^*}^c_{u,s}))$ represent the expected SAVG utility acquired in the future from the display units in $S_{\text{fut}}(c, s, {x^*}^c_{u,s})$. To strike a balance between the increment of SAVG utility in the current iteration and in the future, AVG-D examines every $\alpha = {x^*}^c_{u,s}$ to maximize

$$\begin{aligned}f(c, s, {x^*}^c_{u,s}) = {} & \text{ALG}(S_{\text{tar}}(c, s, {x^*}^c_{u,s})) \\ & + r \cdot \text{OPT}_{\text{LP}}(S_{\text{fut}}(c, s, {x^*}^c_{u,s})),\end{aligned}$$

where

$$\text{ALG}(S_{\text{tar}}(c, s, {x^*}^c_{u,s})) = \sum_{(\hat{u},s) \in S_{\text{tar}}} \text{p}(\hat{u}, c) + \sum_{\substack{e=(\hat{u},\hat{v}) \\ (\hat{u},s),(\hat{v},s) \in S_{\text{tar}}}} \text{w}^c_e,$$

$$\begin{aligned}\text{OPT}_{\text{LP}}(S_{\text{fut}}(c, s, {x^*}^c_{u,s})) = {} & \sum_{c \in \mathcal{C}} \sum_{s=1}^{k} [\sum_{(\hat{u},\hat{s}) \in S_{\text{fut}}} \text{p}(\hat{u}, c) {x^*}^c_{\hat{u},\hat{s}} \\ & + \sum_{\substack{e=(\hat{u},\hat{v}) \\ (\hat{u},\hat{s}),(\hat{v},\hat{s}) \in S_{\text{fut}}}} \text{w}^c_e {y^*}^c_{e,\hat{s}}],\end{aligned}$$

$\text{w}^c_e = \tau(\hat{u}, \hat{v}, c) + \tau(\hat{v}, \hat{u}, c)$ for all $e = (\hat{u}, \hat{v}) \in E$, and r is the balancing ratio. In other words, AVG-D simultaneously optimizes the current and potential future SAVG utility in each iteration.

[8] The current best time complexity for solving linear program equals the complexity of matrix multiplication, or roughly $O(N^{2.5})$ for N variables [16]. However, practical computation generally takes much less time.

Algorithm 3 Deterministic Alignment-aware VR Subgroup Formation

Input: X^*
Output: An SAVG k-Configuration $\mathbf{A}$

$\mathbf{A}(\hat{u},\hat{s}) \leftarrow$ NULL for all $\hat{u},\hat{s}$
$X^* \leftarrow X^*_{\mathrm{LP}}$
while some entry in $\mathbf{A}$ is NULL **do**
  $c,s,u \leftarrow \arg\max_{c,s,u} f(c,s,{x^*}^c_{u,s})$
  $\alpha \leftarrow {x^*}^c_{u,s}$
  for $\hat{u} \in V$ **do**
    if $\mathbf{A}(\hat{u},s) =$ NULL and $\mathbf{A}(\hat{u},t) \neq c \forall t \neq s$ **then**
      if ${x^*}^c_{\hat{u},s} \geq \alpha$ **then**
        $\mathbf{A}(\hat{u},s) \leftarrow c$
return A

Table 6: SAVG k-Configuration returned by AVG for Example 4.

	Slot 1	Slot 2	Slot 3
u_A	c_5	c_2	c_1
u_B	c_2	c_4	c_1
u_C	c_3	c_4	c_5
u_D	c_5	c_4	c_1

It is summarized in Algorithm 3.

Example 5. For the instance in Example 2, the first iteration of AVG-D with $r = \frac{1}{4}$ sets $\alpha = {x^*}^{c_5}_{u_B,1} = 0$, and CSF with $(c,s,{x^*}^c_{u,s}) = (c_5,1,0)$ co-displays the SP camera to everyone at slot 1, where $S_{\mathrm{tar}}(c,s,{x^*}^c_{u,s}) = \{(u_A,1),(u_B,1),(u_C,1),(u_D,1)\}$, $S_{\mathrm{fut}}(c,s,{x^*}^c_{u,s}) = \{(u_A,2),(u_B,2),(u_C,2),(u_D,2),(u_A,3),(u_B,3),(u_C,3),(u_D,3)\}$, $\mathrm{ALG}(S_{\mathrm{tar}}(c,s,{x^*}^c_{u,s})) = \mathrm{p}(u_A,c_5) + \mathrm{p}(u_B,c_5) + \mathrm{p}(u_C,c_5) + \mathrm{p}(u_D,c_5) + \mathrm{w}^{c_5}_{(u_A,u_B)} + \mathrm{w}^{c_4}_{(u_A,u_C)} + \mathrm{w}^{c_4}_{(u_A,u_D)} + \mathrm{w}^{c_4}_{(u_B,u_C)} = 3.35$, $\mathrm{OPT}_{\mathrm{LP}}(S_{\mathrm{fut}}(c,s,{x^*}^c_{u,s})) = 6.97$, and $f(c,s,{x^*}^c_{u,s}) = 3.35 + \frac{1}{4}\cdot 6.97 = 5.09$ is maximized. The next iteration selects $\alpha = {x^*}^{c_1}_{u_A,2} = 0.33$ to co-display the tripod to {Alice, Bob, Dave} at slot 2. AVG-D selects $\alpha = {x^*}^{c_2}_{u_D,3} = 0$, $\alpha = {x^*}^{c_4}_{u_A,3} = 0$, and $\alpha = {x^*}^{c_3}_{u_D,2} = 0$ in the next three iterations, resulting in a total SAVG utility of 9.85, which is slightly larger than AVG (9.75) in Example 4. The returned SAVG k-Configuration is shown in Table 7.

For this running example, the optimal solution is the SAVG k-Configuration shown at the top of Figure 1(a), with a total SAVG utility of 10.35. The personalized approach retrieves the top-3 preferred items for each user, e.g., $\langle c_5,c_2,c_1\rangle$ for Alice, and achieves a total SAVG utility of 8.25. For the group approach, the total SAVG utility for the universal subgroup {Alice, Bob, Charlie, Dave} viewing each item is aggregated. For example, the total SAVG utility of the subgroup of all users viewing c_1 would be $\sum_{u\in V} \mathrm{p}(u,c_1) + \sum_{u,u' in V} \tau(u,u',c_1)$, which equals the summation of all values in the first row in Table 1, or 2.6 (note that $\lambda = 0.5$ and the objective is scaled up by 2 here). The top-3 items, i.e., $\langle c_5,c_1,c_2\rangle$, is retrieved and co-displayed to all users. This achieves a total SAVG utility of 8.35.

We also compare with two variations of the subgroup approach, namely the subgroup-by-friendship and subgroup-by-preference approaches, where the former pre-partitions the subgroups based on social relations among the users, while the latter finds subgroups with similar item preferences. The subgroup-by-friendship approach first partitions the social network into two equally sized and dense subgroups {Alice, Dave} and {Bob, Charlie}. Items are then determined analogously to the group approach, resulting in a total SAVG utility of 8.4. Finally, the subgroup-by-preference approach partitions the network into {Alice, Bob} and {Charlie, Dave}. The total SAVG utility would be 8.7.

The configurations returned by these four baseline approaches are consistent with Figure 1 and are also summarized in Table 8. To sum up, the total SAVG utility for AVG, AVG-D, and the personalized, group, subgroup-by-friendship and subgroup-by-preference approaches, are 9.75, 9.85, 8.25, 8.35, 8.4 and 8.7, respectively. Therefore, both AVG and AVG-D achieves a near-optimal solution. □

Table 7: SAVG k-Configuration returned by AVG-D for Example 4.

	Slot 1	Slot 2	Slot 3
u_A	c_5	c_1	c_2
u_B	c_5	c_1	c_2
u_C	c_5	c_3	c_2
u_D	c_5	c_1	c_4

Table 8: Configuration by the baseline approaches for Example 4.

personalized	Slot 1	Slot 2	Slot 3
u_A	c_5	c_2	c_1
u_B	c_2	c_1	c_4
u_C	c_3	c_4	c_2
u_D	c_4	c_5	c_3
group			
$\{u_A,u_B,u_C,u_D\}$	c_5	c_1	c_2
subgroup-by-friendship			
$\{u_A,u_D\}$	c_5	c_1	c_4
$\{u_B,u_C\}$	c_2	c_4	c_3
subgroup-by-preference			
$\{u_A,u_B\}$	c_2	c_1	c_5
$\{u_C,u_D\}$	c_4	c_5	c_3

In AVG-D, r serves as a turning knob to strike a good balance between the current increment in SAVG utility and the potential SAVG utility in the future. Specifically, we prove that by setting $r = \frac{1}{4}$ in each iteration, AVG-D is a deterministic 4-approximation algorithm for SVGIC.

Theorem 5. Given the optimal fractional solution X^*, AVG-D returns a worst-case 4-approximate SAVG k-Configuration in $O(n \cdot m \cdot k \cdot |E|)$-time.

In DPS, let $\mathrm{OPT}_{\mathrm{LP}}(S_{\mathrm{cur}})$ denote the total SAVG utility from only the slots in S_{cur} according to X^*. More specifically,

$$\mathrm{OPT_{LP}}(S_{\mathrm{cur}}) = \sum_{\hat{c}\in\mathcal{C}}\sum_{\hat{s}=1}^{k} \sum_{\substack{(\hat{u},\hat{s})\in\\ S_{\mathrm{cur}}(c,s,{x^*}^c_{u,s})}} (\mathrm{p}(\hat{u},\hat{c}){x^*}^{\hat{c}}_{\hat{u},\hat{s}} + \sum_{\substack{e=(\hat{u},\hat{v})\\ (\hat{v},\hat{s})\in S_{\mathrm{cur}}(c,s,{x^*}^c_{u,s})}} \mathrm{w}^{\hat{c}}_e {y^*}^{\hat{c}}_{e,\hat{s}}).$$

Similar to above, an *iteration* for AVG-D refers to DPS selecting $(c,s,{x^*}^c_{u,s})$ and the subsequent CSF. We first outline the proof sketch. To prove the approximation ratio for AVG-D, we first need to show that, in an iteration of AVG-D, *if* $(c,s,{x^*}^c_{u,s})$ *is selected with RPS instead of DPS*, so that the probability of selecting $(c,s,{x^*}^c_{u,s})$ is the sum of selecting all (c,s,α) that lead to an equivalent outcome as selecting $(c,s,{x^*}^c_{u,s})$, then the expected value of $f(c,s,{x^*}^c_{u,s})$ is at least $\frac{\mathrm{OPT_{LP}}(S_{\mathrm{cur}})}{4}$. Thus, for each iteration there exists at least one (c,u,s) with $f(c,s,{x^*}^c_{u,s}) \geq \frac{\mathrm{OPT_{LP}}(S_{\mathrm{cur}})}{4}$, and AVG-D is therefore a 4-approximating algorithm.

We begin with proving that in every iteration t, if (c,s,α) is selected with RPS instead of DPS, $\mathbb{E}[f(c,s,\alpha)] \geq \frac{1}{4}\cdot \mathrm{OPT_{LP}}(S_{\mathrm{cur}})$, where the function $f(c,s,\alpha)$ is the evaluation criteria in DPS. Therefore, $f(c,s,{x^*}^c_{u,s})$ in DPS is also at least $\frac{1}{4}\cdot\mathrm{OPT_{LP}}(S_{\mathrm{cur}})$, since DPS maximizes $f(c,s,{x^*}^c_{u,s})$.

Lemma 6. In any iteration of AVG, if (c,s,α) is selected by RPS, $\mathbb{E}[f(c,s,\alpha)] \geq \frac{1}{4}\cdot\mathrm{OPT_{LP}}(S_{\mathrm{cur}})$.

Proof. In any iteration t, for each $e=(\hat{u},\hat{v})$, c, and s with $(\hat{u},s)$ and $(\hat{v},s)$ in S_{cur} (i.e., $\hat{u}$ and $\hat{v}$ are eligible for (c,s)), by Lemma 4, the probability that $\hat{u}$ and $\hat{v}$ are co-displayed c at slot s is

$$\frac{\min\{{x^*}^c_{\hat{u},s},{x^*}^c_{\hat{v},s}\}}{k\cdot m} = \frac{{y^*}^c_{e,s}}{k\cdot m}.$$

Therefore,

$$\begin{aligned}
&\mathbb{E}[\mathrm{ALG}(S_{\mathrm{tar}}(c,s,\alpha))]\\
=&\mathbb{E}[\sum_{(\hat{u},s)\in S_{\mathrm{tar}}(c,s,\alpha)} \big(\mathrm{p}(\hat{u},c) + \sum_{(\hat{v},s)\in S_{\mathrm{tar}}(c,s,\alpha)} \tau(\hat{u},\hat{v},c)\big)]\\
=&\sum_{c\in\mathcal{C}}\sum_{s=1}^{k}\big(\sum_{(\hat{u},s)\in S_{\mathrm{cur}}} \mathrm{p}(\hat{u},c)\frac{{x^*}^c_{\hat{u},s}}{k\cdot m} + \sum_{(\hat{u},s),(\hat{v},s)\in S_{\mathrm{cur}}} \mathrm{w}^c_e\frac{{y^*}^c_{e,s}}{k\cdot m}\big)\\
=&\frac{1}{k\cdot m}\sum_{c\in\mathcal{C}}\sum_{s=1}^{k}\big(\sum_{(\hat{u},s)\in S_{\mathrm{cur}}} \mathrm{p}(\hat{u},c){x^*}^c_{\hat{u},s} + \sum_{\substack{e=(\hat{u},\hat{v})(\hat{u},s),\\ (\hat{v},s)\in S_{\mathrm{cur}}}} \mathrm{w}^c_e{y^*}^c_{e,s}\big)\\
=&\frac{\mathrm{OPT_{LP}}(S_{\mathrm{cur}})}{k\cdot m}.
\end{aligned}$$

Next, from Lemma 3 and 5, for each $e=(\hat{u},\hat{v})$, $\hat{c}$, and $\hat{s}$ such that $(\hat{u},\hat{s}),(\hat{v},\hat{s})\in S_{\mathrm{cur}}$, the probability that $(\hat{u},\hat{s})\in S_{\mathrm{fut}}(c,s,\alpha)$ is at least $1-P^{\hat{u}}_{\mathrm{ne}}$, and the probability that $(\hat{u},\hat{s}),(\hat{v},\hat{s})\in S_{\mathrm{fut}}(c,s,\alpha)$ is at least $1-P^{e}_{\mathrm{ne}}$, which implies the probability that both $\hat{u}$ and $\hat{v}$ are eligible for $(\hat{c},\hat{s})$ in iteration $(t+1)$ is at least $1-\frac{4}{k\cdot m}$. Therefore, the expected contribution of the social utility from the co-display of $\hat{c}$ to $\hat{u}$ and $\hat{v}$ at slot $\hat{s}$ is at least $\mathrm{w}^{\hat{c}}_e\cdot(1-\frac{4}{k\cdot m})$. Thus,

$$\begin{aligned}
&\mathbb{E}[\mathrm{OPT_{LP}}(S_{\mathrm{fut}}(c,s,\alpha))]\\
\geq&\sum_{\hat{c}\in\mathcal{C}}\sum_{\hat{s}=1}^{k}[\sum_{(\hat{u},\hat{s})\in S_{\mathrm{cur}}}(1-P^{\hat{u}}_{\mathrm{ne}})\mathrm{p}^{\hat{c}}_{\hat{u},\hat{s}}{x^*}^{\hat{c}}_{\hat{u},\hat{s}} + \sum_{\substack{(\hat{u},\hat{s}),\\ (\hat{v},\hat{s})\in S_{\mathrm{cur}}}}(1-P^{e}_{\mathrm{ne}})\mathrm{w}^{\hat{c}}_e{y^*}^{\hat{c}}_{e,\hat{s}}]\\
\geq&(1-\frac{4}{k\cdot m})\cdot\sum_{\hat{c}\in\mathcal{C}}\sum_{\hat{s}=1}^{k}[\sum_{(\hat{u},\hat{s})\in S_{\mathrm{cur}}}\mathrm{p}^{\hat{c}}_{\hat{u},\hat{s}}{x^*}^{\hat{c}}_{\hat{u},\hat{s}} + \sum_{\substack{(\hat{u},\hat{s}),\\ (\hat{v},\hat{s})\in S_{\mathrm{cur}}}}\mathrm{w}^{\hat{c}}_e{y^*}^{\hat{c}}_{e,\hat{s}}]\\
=&(1-\frac{4}{k\cdot m})\mathrm{OPT_{LP}}(S_{\mathrm{cur}}).
\end{aligned}$$

Combining the above, we have

$$\begin{aligned}
\mathbb{E}[f(c,s,\alpha)] &= \mathbb{E}[\mathrm{ALG}(S_{\mathrm{tar}}(c,s,\alpha))] + \frac{1}{4}\cdot\mathbb{E}[\mathrm{OPT_{LP}}(S_{\mathrm{fut}}(c,s,\alpha))]\\
&\geq \frac{\mathrm{OPT_{LP}}(S_{\mathrm{cur}})}{k\cdot m} + \frac{1}{4}\cdot\mathrm{OPT_{LP}}(S_{\mathrm{cur}})\cdot(1-\frac{4}{k\cdot m})\\
&= \frac{\mathrm{OPT_{LP}}(S_{\mathrm{cur}})}{4}.
\end{aligned}$$

The lemma follows. □

Back to AVG-D, let the total number of iterations be T. Let S^t_{tar} and S^t_{fut} denote $S_{\mathrm{tar}}(c,s,{x^*}^c_{u,s})$ and $S_{\mathrm{fut}}(c,s,{x^*}^c_{u,s})$ in the t-th iteration of AVG-D, and $S^t_{\mathrm{cur}} = S^t_{\mathrm{tar}}\cup S^t_{\mathrm{fut}}$. By Lemma 6, $f(c,s,{x^*}^c_{u,s}) = \max_{\hat{c},\hat{s},\hat{u}} f(\hat{c},\hat{s},{x^*}^{\hat{c}}_{\hat{u},\hat{s}}) \geq \mathbb{E}[f(c,s,\alpha)] \geq \frac{\mathrm{OPT_{LP}}(S_{\mathrm{cur}})}{4}$ for every iteration t in AVG-D. Therefore, since the solution of AVG-D performs no worse than the randomized approach in the above lemma,

$$\begin{aligned}
&\mathrm{ALG}(S^1_{\mathrm{tar}}) + \frac{1}{4}\cdot\mathrm{OPT_{LP}}(S^1_{\mathrm{fut}}) \geq \frac{1}{4}\cdot\mathrm{OPT_{LP}}(S^1_{\mathrm{cur}})\\
&\mathrm{ALG}(S^2_{\mathrm{tar}}) + \frac{1}{4}\cdot\mathrm{OPT_{LP}}(S^2_{\mathrm{fut}}) \geq \frac{1}{4}\cdot\mathrm{OPT_{LP}}(S^2_{\mathrm{cur}}) = \frac{1}{4}\cdot\mathrm{OPT_{LP}}(S^1_{\mathrm{fut}})\\
&\qquad\vdots\\
&\mathrm{ALG}(S^T_{\mathrm{tar}}) + \frac{1}{4}\cdot\mathrm{OPT_{LP}}(S^T_{\mathrm{fut}}) \geq \frac{1}{4}\cdot\mathrm{OPT_{LP}}(S^{T-1}_{\mathrm{fut}}).
\end{aligned}$$

By summing up all the above inequalities, we have

$$\begin{aligned}
\sum_{t=1}^{T}\mathrm{ALG}(S^t_{\mathrm{tar}}) &\geq \frac{\mathrm{OPT_{LP}}(S^1_{\mathrm{cur}})}{4} - \frac{\mathrm{OPT_{LP}}(S^T_{\mathrm{fut}})}{4}\\
&= \frac{\mathrm{OPT}}{4} - 0 = \frac{\mathrm{OPT}}{4},
\end{aligned}$$

which proves the approximation ratio.

Similar to that in AVG, a slight modification on AVG-D is to select only $(c,s,{x^*}^c_{u,s})$ that leads to a nonempty target group, so that $T=O(nk)$. We first prove that there does exist such $(c,s,{x^*}^c_{u,s})$ that $f(c,s,{x^*}^c_{u,s}) \geq \frac{1}{4}\cdot\mathrm{OPT_{LP}}(S_{\mathrm{cur}}(c,s,{x^*}^c_{u,s}))$. Consider those cases where $\alpha = {x^*}^c_{u,s}$ is too large that the target group is empty. In this case $S_{\mathrm{cur}}(c,s,{x^*}^c_{u,s}) = S_{\mathrm{fut}}(c,s,{x^*}^c_{u,s})$, which implies $f(c,s,{x^*}^c_{u,s}) = \frac{1}{4}\cdot\mathrm{OPT_{LP}}(S_{\mathrm{cur}}(c,s,{x^*}^c_{u,s}))$, i.e., exactly the expected value. Suppose that no other $(c,s,{x^*}^c_{u,s})$ exists so that $f(c,s,{x^*}^c_{u,s}) \geq \frac{1}{4}\cdot\mathrm{OPT_{LP}}(S_{\mathrm{cur}}(c,s,{x^*}^c_{u,s}))$, then the expected value should be lower than $\frac{1}{4}\cdot\mathrm{OPT_{LP}}(S_{\mathrm{cur}}(c,s,{x^*}^c_{u,s}))$, a contradiction. Thus, such

$(c, s, x^{*c}_{u,s})$ must exist, and AVG-D can always choose such $(c, s, x^{*c}_{u,s})$.

The number of iterations is thus also $O(nk)$. For each iteration, there are $O(nmk)$ combinations of pivot parameters. Finding $S_{\text{tar}}(c, s, x^{*c}_{u,s})$ and S_{fut}, as well as executing CSF, require $O(n)$ time. To find the optimal pivot parameters in DPS in each iteration, a simple approach is to examine all $O(nmk)$ combinations of pivot parameters. For each $(c, s, x^{*c}_{u,s})$, finding $f(c, s, x^{*c}_{u,s})$ requires $O(E)$ time. Thus, each iteration needs $O(nmk|E|)$ time. Since $T = O(nk)$, the total time complexity of AVG-D, including the config phase, is therefore $O(\text{LP}) + O(n^2mk^2|E|)$. By using heaps to store the utility factors and reordering the computation, repeated calculations can be avoided to reduce the complexity to $O(\text{LP}) + O(nmk^2|E|)$. Alternatively, effectively parallelizing the computation of $f(c, s, x^{*c}_{u,s})$ can achieve at most a speedup factor of nmk, which further lowers the time complexity to $O(\text{LP}) + O(nmk|E|)$. □

4.4 Extensions of AVG

In the following, we show that AVG (and AVG-D) supports values of $\lambda \neq \frac{1}{2}$ via a simple scaling on the inputs. We then extend them to support SVGIC-ST by tailoring CSF with consideration of the additional VR-related constraints.

Other values of λ. To support the cases with $\lambda \neq \frac{1}{2}$, AVG first sets the *scaled preference values* for each (u, c) as $\text{p'}(u, c) = \frac{1-\lambda}{\lambda}\text{p}(u, c)$. This value is then used instead of $\text{p}(u, c)$ in the LP relaxation problem, as well as in the computation of $f(c, s, x^{*c}_{u,s})$ in DPS in AVG-D. Since

$$
\begin{aligned}
&(1-\lambda)\cdot \text{p}(u,c) + \lambda \cdot \sum_{v \mid u \overset{c}{\longleftrightarrow} v} \tau(u,v,c) \\
=&2\lambda \cdot (\frac{1}{2}\cdot \text{p'}(u,c) + \frac{1}{2}\cdot \sum_{v \mid u \overset{c}{\longleftrightarrow} v} \tau(u,v,c)),
\end{aligned}
$$

each instance with $\lambda \neq \frac{1}{2}$ can be transformed to an instance with $\lambda = \frac{1}{2}$, in which AVG can achieve the approximation ratio, by this scaling of input parameters.

Extending AVG for SVGIC-ST. To modify AVG (and consequently, AVG-D) to support the teleportation and subgroup size constraints in VR, the LP relaxation problem is first replaced by the corresponding formulation in Section 3.2, i.e., with Constraints 8, 9 and 10. Next, in each iteration of CSF, instead of displaying the focal item c to all eligible users $\hat{u}$ with $x^{*c}_{u,s} \geq \alpha$, CSF first checks the number of users already displayed c at the focal slot s. It then iteratively adds *the eligible user with the highest utility factor of c at slot s* until the number of users displayed c at slot s reaches the size constraint M or every user is examined. If the size constraint is reached, CSF *locks* the item c at slot s by setting $x^{*c}_{u,s} = 0$ for all remaining eligible users, as well as removes (c, s, α) from the set of candidate parameter sets in AVG-D. Therefore, AVG never displays more than M users to the same item in the final SAVG k-Configuration.

5. EXPERIMENTS

In this section, we evaluate the proposed AVG and AVG-D along with various baseline algorithms on three real datasets. Moreover, a case study on the real dataset is provided to demonstrate the characteristics of solutions derived from different approaches. Finally, we build a prototype of a VR store with Unity and SteamVR to conduct a user study.

5.1 Experiment Setup and Evaluation Plan

Datasets. To evaluate the proposed algorithms, three real datasets are tested in the experiment. The first dataset *Timik* [25] is a 3D VR social network containing 850k vertices (users) and 12M edges (friendships) with 12M check-in histories of 849k virtual Point-of-Interests (POIs). The second dataset, *Epinions* [1], is a website containing the reviews of 139K products and a social trust network with 132K users and 841K edges. The third dataset, *Yelp* [6], is a location-based social network (LBSN) containing 1.5M vertices, 10M edges, and 6M check-ins. For Timik and Yelp, we follow the settings in [9, 20, 45] to treat POIs in the above datasets as the candidate items in SVGIC. The preference utility and social utility values are learned by the PIERT learning framework [31] which jointly models the social influence between users and the latent topics of items. Following the scales of the experiments in previous research [42, 44], the default number of slots k, the default number of items m, and the default size of user set n selected from the social networks to visit a VR store are set as 50, 10000, and 125, respectively.[9]

Baseline Methods. We compare AVG and AVG-D with five baseline algorithms: Personalized Top-k (PER), Fairness Maximization in Group recommendation (FMG) [44], Social-aware Diverse and Preference selection (SDP) [45], Group Recommendation and Formation (GRF) [42], and Integer Programming (IP). PER and FMG correspond to the two baseline approaches outlined in Section 1. Specifically, PER retrieves the top-k preferred items for each user (the personalized approach), while FMG selects a bundled itemset for all users as a group (the group approach) with considerations of fairness of preference among the users. SDP selects socially-tight subgroups to display their preferred items, which corresponds to the subgroup approach outlined in Section 4. GRF splits the input users into subgroups with similar item preferences without considering the social network topology, which can be viewed as a variation of the subgroup approach where the subgroups are partitioned based on preferences instead of social connections. Finally, IP is the integer program formulated in Section 3.2 that finds the optimal solutions of small SVGIC instances by Gurobi [2]. All algorithms are implemented in an HP DL580 Gen 9 server with four 3.0 GHz Intel CPUs and 1TB RAM. Each result is averaged over 50 samples.

Evaluation Metrics.
To evaluate the algorithms considered for SVGIC and analyze the returned SAVG k-Configurations, we introduce the following metrics: 1) total SAVG utility achieved, 2) total execution time (in seconds), 3) the percentages of personal preference utility (*Personal%*) and social utility (*Social%*) in total SAVG utility, 4) the percentage of *Inter*-subgroup edges (*Inter%*) and *Intra*-subgroup edges (*Intra%*) in the returned partition of subgroups, 5) the average network density among partitioned subgroups, normalized by the average density of the original social network, 6) the percentage of friend pairs viewing common items together (*Co-*

[9]Note that a small-to-medium scale value of n is practical for VR applications, e.g., VRChat [5] allows at most 16 users, and IrisVR [4] allows up to 12 users.

display%), 7) the percentage of users viewing items alone (*Alone%*), and 8) *regret ratio*. A game-theory based measurement for satisfaction of individual users and the overall *fairness* of the solution, the regret ratio reg(u) [30], is defined as follows.

$$\mathrm{reg}(u) \equiv 1 - \frac{\sum_{c \in \mathbf{A}^*(u,\cdot)} \mathrm{w}_{\mathbf{A}}(u,c)}{\max_{\mathcal{C}_u} \sum_{c \in \mathcal{C}_u} \bar{\mathrm{w}}_{\mathbf{A}}(u,c)}$$

where the numerator is the achieved SAVG utility, the denominator with $\bar{\mathrm{w}}_{\mathbf{A}}(u,c) = (1-\lambda)\cdot\mathrm{p}(u,c)+\lambda\cdot\sum_{v\in V}\tau(u,v,c)$ is an *upper bound* of possible SAVG utility when all users view c together with user u, and $\mathcal{C}_u$ is a k-itemset, corresponding to a very optimistic scenario favoring u the most. Note that the second term of $\bar{\mathrm{w}}_{\mathbf{A}}(u,c)$ is different from $\mathrm{w}_{\mathbf{A}}(u,c)$ in Definition 3. In other words, the upper bound is the SAVG utility of u if she dictates the whole SAVG k-Configuration selfishly. Therefore, a low reg(u) implies that user u is relatively satisfied with the SAVG k-Configuration, and *fairness* among the users can be observed by inspecting the distribution of regret ratio in the final configurations. Moreover, for evaluating the performance of all methods in SVGIC-ST, we measure 9) *feasibility ratio*: the number of feasible solutions divided by the total number of instances, and 10) *group size constraint violation*: the number of partitioned subgroups exceeding the predefined size constraint.

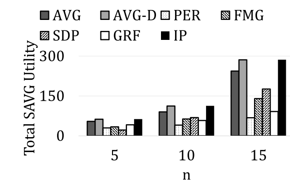
User Study. Finally, we build a prototype of VR store with Unity 2017.1.1.1 (64bit), Photon Unity Network free 1.86, SteamVR Plugin 1.2.2, VRTK, and 3ds Max 2016 for hTC VIVE HMD. We detail the user study in Section 5.7.

5.2 Comparisons on Small Datasets

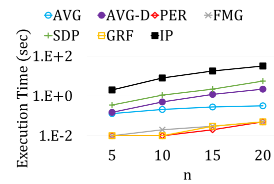
To evaluate the performance of the above algorithms, we use IP to derive the optimal total SAVG utility on small datasets.[10] The social networks and items in the small datasets are respectively sampled by random walk and uniform sampling from Timik according to the setting of [37]. Figure 2(a) manifests that AVG and AVG-D outperform the other approaches regarding the total SAVG utility for different n (i.e., the size of user set). Solutions of AVG-D are close to optimal since AVG-D extracts the target subgroups by jointly optimizing the current and expected future total SAVG utility in each iteration. AVG-D outperforms other baselines by at least 50.8% to 62.8% as n grows, because the social interactions on extracted target subgroups become increasingly important when the size of user set increases. In contrast, the values of total SAVG utility of PER grows slowly because it is not designed to cope with the interplay and trade-off between personal preferences and social interactions. Moreover, while FMG and GRF seem to benefit from large values of n, their solution qualities are still limited due to the fixed partition of subgroups generated without leveraging CID.

Figure 2(c) presents the total SAVG utility under varied numbers of items (m). However, m does not seem to affect the total SAVG utility by much since any user's top preferred items are already contained in the top-100 items. Moreover, Figure 2(e) presents the total SAVG utility under varied number of slots (k). AVG-D and AVG significantly outperform the baselines by at least 134.7% and 102.1% in

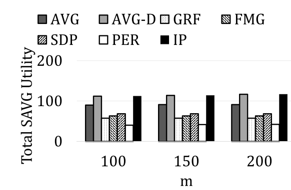
[10]Since solving IP is NP-hard, Gurobi cannot solve large instances within hours.



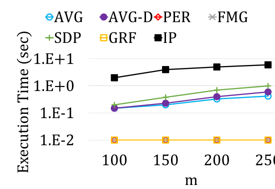
(a) Total SAVG utility vs. size of user set (n).



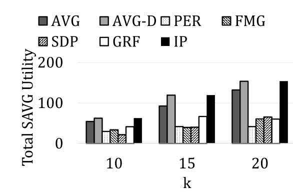
(b) Execution time vs. size of user set (n).



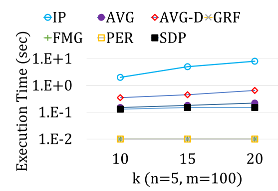
(c) Total SAVG utility vs. size of item set (m).



(d) Execution time vs. size of item set (m).



(e) Total SAVG utility vs. number of slots (k).



(f) Execution time vs. number of slots (k).

Figure 2: Comparisons on small datasets.

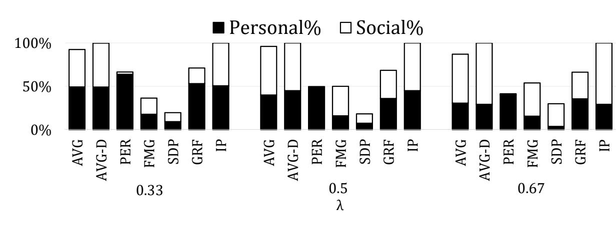


Figure 3: Normalized total SAVG utility of diff. λ.

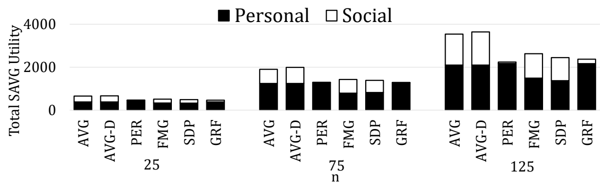


Figure 4: Total SAVG utility vs. size of user set (n).

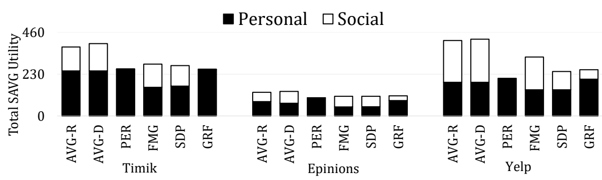


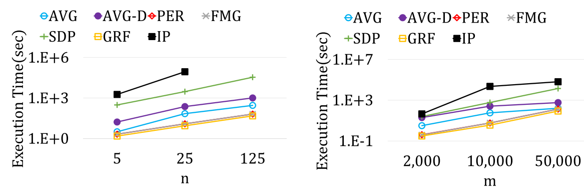
Figure 5: Total SAVG utility in diff. datasets.

terms of the total SAVG utility when k grows to 20, because CSF leverages the flexibility provided by CID to optimize the social utility for different slots, which is beneficial for a large k as it becomes difficult to find more commonly interested items for static subgroup members. Figures 2(b), 2(d) and 2(f) show the execution time by varying the size of user set (n), the size of item set (m), and the number of slots (k). The running times of AVG and AVG-D are at most 7.5% and 17.4% of that of IP. AVG and AVG-D require slightly more time than PER, GRF, and FMG, since the baseline approaches focus only on one of preference utility, social utility, or subgroup partition, instead of considering all of them jointly. Figure 3 evaluates the impact of λ on the total SAVG utility of all schemes normalized by that of IP. The normalized total SAVG utilities of FMG and SDP improve when the social utility becomes more important as λ grows. However, it is difficult for them to address the diverse personal preferences. In contrast, PER achieves the highest preference utility and the lowest social utility, but tends to generate a small total SAVG utility.

5.3 Sensitivity and Scalability Tests on Large Datasets

Next, we evaluate the efficiency and efficacy of AVG and AVG-D in large datasets with the scales of the three dimensions following previous research [42, 44], i.e., $m = 10000$, $k = 50$, and $n = 125$. Figure 4 presents the total SAVG utility by varying the sizes of user set in Timik. The results manifest that AVG and AVG-D outperform all baselines by at least 30.1%, while AVG-D is slightly better than AVG since it selects better pivot parameters for CSF. Compared with GRF, the improvement of AVG-D grows from 43.6% to 54.6% as n increases, since GRF splits the users into subgroups without considering social relations, but social interactions among *close friends* become more important for a larger group. By striking a balance between preference and social utility, AVG and AVG-D achieve a greater total SAVG utility. Compared with PER and GRF, FMG achieves a higher social utility but a lower preference utility because it displays a universal configuration to all users.

Figures 5 compares the results on Timik, Yelp, and Epinions. The social utility obtained in Epinions is lower than in Yelp due to the sparser social relations in the review network, and group consensus thereby plays a more important



(a) Execution time vs. size of user set (n).

(b) Execution time vs. size of item set (m).

Figure 6: Execution time in Yelp.

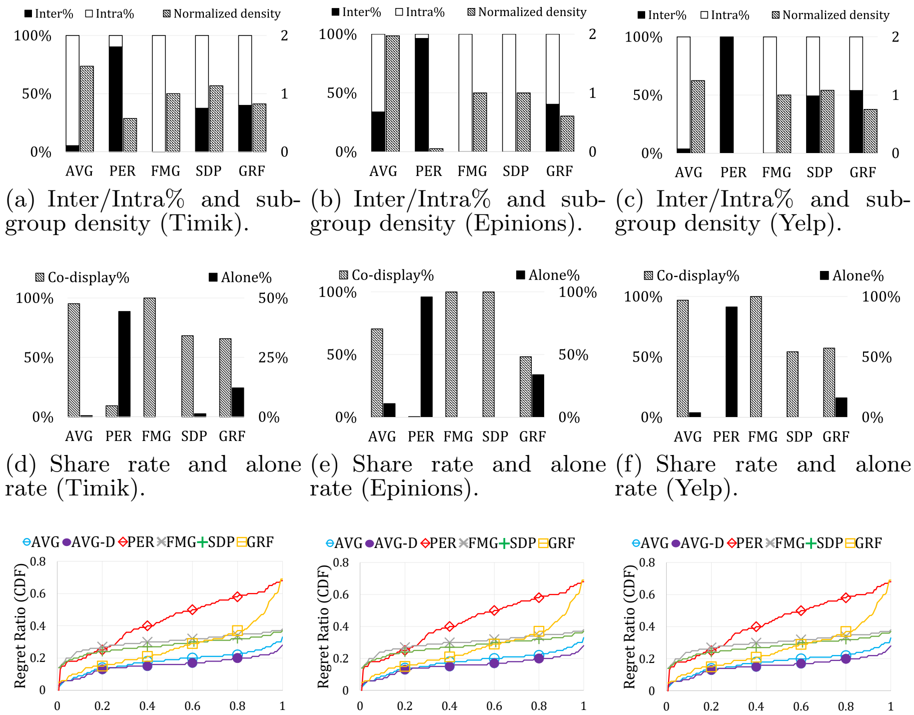
role in Yelp. Despite the different characteristics of datasets, AVG and AVG-D prevail in all datasets and outperform all baselines since CSF operates on the utility factors from the optimal LP solution, which does strike a good balance among all factors. FMG and SDP benefit from the high social utility in Yelp and outperform PER. By contrast, PER performs nearly as good as FMG and SDP in Epinions since the social utility is lower.

Figure 6(a) presents the execution time in Yelp with different n. IP cannot terminate within 24 hours when $n \geq 25$, and SDP needs 300 seconds to return a solution even when $n = 5$. Figure 6(b) shows the execution time with different m. Note that AVG and AVG-D are both more scalable to m than the baseline approaches because CSF exploits the fractional solution without m in the complexity. Although AVG-D provides a stronger theoretical guarantee, the scalability of AVG on n is better than AVG-D because AVG samples the target subgroups randomly. However, practical VR applications, e.g., VRChat [5] and IrisVR [4], seldom have $n > 25$. Moreover, recall that the SVGIC problem simultaneously explores three dimensions: n (the size of user set), m (the size of item set), and k (the number of slots). Therefore, forming the subgroups and configuring the items in SVGIC is combinatorially harder than traditional optimization problems in group recommendation [42, 44] or group formation [46, 47].

5.4 Comparisons on Subgroup Metrics

In the following, we analyze the subgroups in the SAVG k-Configurations returned from all algorithms[11] in terms of subgroup-related metrics. Figures 7(a), 7(b), and 7(c) compare the ratios of Inter-/Intra-subgroup edges averaged across all slots in the SAVG k-Configurations in the Timik, Yelp, and Epinions datasets, respectively. It also shows the average network density among the partitioned subgroups normalized by the original density of the input social network. The results from all datasets indicate that the majority of preserved edges by AVG are within the same subgroups (large Intra%). FMG achieves 100% in Intra%, 0% in Inter%, and 1 in normalized density because it consistently views the whole network as a large subgroup. In contrast, PER has a high Inter% as it separates most users into independent subgroups to display their favorite items. This phenomenon is more obvious in Yelp (with a 100% Inter%, i.e., *every* user is left alone as a tiny subgroup) than in Timik (about 30% of the social edges are Intra-subgroup edges) or Epinions (a nonzero Intra%). This is because Yelp is a product-review application, where the POIs (restaurants, stores, service centers, etc.) are highly diversified, making it

[11]Some of the results of AVG-D are omitted as they are similar to AVG.



(a) Inter/Intra% and subgroup density (Timik). (b) Inter/Intra% and subgroup density (Epinions). (c) Inter/Intra% and subgroup density (Yelp).

(d) Share rate and alone rate (Timik). (e) Share rate and alone rate (Epinions). (f) Share rate and alone rate (Yelp).

(g) Regret ratio (Timik). (h) Regret ratio (Epinions). (i) Regret ratio (Yelp).

Figure 7: Comparisons on subgroup metrics.

more difficult for different users to have aligned preferences, e.g., user A has her k-th favorite POI (indicated by check-in records) identical to user B's k-th favorite POI, such that PER happens to co-display the POI to them. On the other hand, there exist a small subset of widely liked or adopted items in Epinions that appear as most users' favorite (hence the small nonzero Intra% of PER), while famous VR locations (e.g., transportation hubs in VR world of Timik) are inclined to be associated with high preference utilities by the exploited recommendation learning models as they generate a lot of check-ins among all users. Therefore, they have a higher chance to be co-displayed by PER.

Among all methods, AVG achieves the largest normalized density as CSF carefully considers the utility factors to partition the network into dense communities. The normalized density achieved by AVG in Yelp is the lowest among the three datasets due to the aforementioned issue of diversified interests but is still higher than all other algorithms. As a result, AVG has the most abundant social connections among subgroup members, which usually trigger enthusiastic discussions and purchases. It is worth noting that VR users generally interact with more strangers during the trip, whereas users in traditional LBSN mainly interact with friends in their spatial proximity. Therefore, the local community structures in Timik are less apparent than those of Yelp. As such, despite adopting different partitioning criteria, the average densities of the subgroups retrieved by SDP and GRF do not differ much in Timik.

Figures 7(d), 7(f), and 7(e) illustrate the Co-display% and Alone% among the returned SAVG k-Configurations. As shown, AVG has a high Co-display% near 1.0 (implying almost all pairs of friends are sharing the views of common items) and a near-zero Alone% (showing that almost no users are left alone in the configuration) in all datasets. Note that the Co-display% is based on friend pairs and the Alone% is calculated based on all users, i.e., they are not complementary statistics (thus do not sum up to 100%). Although FMG achieves 100% in Co-display% and 0% in Alone%, it sacrifices the preferences of group members by forming huge subgroups for co-display. On the other hand, both AVG and GRF are able to maintain high values of Co-display% while taking into account the personal preferences. In all datasets, GRF leaves a considerably high portion of users alone because it forms the subgroups according to item preferences. Therefore, some users with unique profiles of interests are more inclined to be left alone. The only other method with a higher Alone% than GRF is PER, which does not facilitate shared views.

Figures 7(g), 7(i), and 7(h) report the CDF of regret ratios of all algorithms in the the Timik, Yelp, and Epinions datasets. AVG and AVG-D consistently have the lowest regret ratios. Among the other approaches, PER incurs the highest regret ratios for users in all datasets, since it does not foster social interactions in shared views. Consistent with the performances on SAVG utility, FMG and SDP outperform PER and GRF in the Timik and Yelp datasets, but their performances are comparable in Epinions because the sparser social relations in the review network generates lower social utility. Interestingly, in Timik and Yelp, some users in GRF are highly satisfied (as the CDF of GRF matches well with that of AVG and AVG-D from the beginning) but some others have very high regret ratios (as the CDF dramatically rises near the end). This indicates that a portion of the users in GRF may actually have their preferences sacrificed, i.e., they are forced to share views on uninterested items with other subgroup members. In contrast, FMG and SDP show flatter CDFs, implying the user preferences are more balanced. However, users in FMG and SDP consistently have regret ratios over 20%, while the regret ratios seldom exceed 20% in AVG/AVG-D; this is because the randomly chosen pivot parameters (in AVG) and the deterministically optimized one (in AVG-D) can effectively form dense subgroups with similar item preferences.

5.5 A Case Study on Partitioning Subgroups

Figure 8 depicts a 2-hop ego network (the subnetwork consisting of all 2-hop friends) of a user A in Yelp and two slots with the highest regret. Note that user A is studied here because she has a unique profile of preference that does not resemble any of her friends (B, E, and F), as shown in the table listing the top-4 POIs of the preference utility for each user. The shaded areas illustrate the partitioned subgroups at the specific slot, while solid and dashed lines depict intra-subgroup and inter-subgroup social edges. The icon beside each subgroup represents the selected POI (displayed item).

As indicated in the table, A has a unique profile of preference that does not resemble any of her friends B, E and F. Therefore, A and her friends are partitioned into distinct subgroups by GRF. On the other hand, AVG assigns a baseball field to $\{A, E, F, G\}$ at slot 1, and a bar to $\{A, B, C, D\}$ at slot 2. Therefore, AVG is able to capture different interests of A and select proper subgroups of friends for them. However, A is not co-displayed the library with $\{B, C, D\}$ at slot 1 and the beach with $\{E, F, G\}$ at slot 2 because none of the two POIs is in A's favor. On the other hand, SDP partitions the ego-network into three cliques based on the network topology and then assigns a music place and a movie theater to $\{A, B\}$, favoring B's but sacrificing A's interests. Moreover, A is inclined to be dissatisfied for not joining $\{E, F, G\}$ to the baseball field (the third preferred POI) at slot 2. GRF gathers the users with similar interests and leaves A alone, since none of her friends is a big fan of the two restaurants. Therefore, the regret ratios for A are 35.2% (SDP), 41.2% (GRF), and 19.6% (AVG) accordingly. Indeed, this case study manifests that flexible partitions of subgroups are crucial the situations of friends not necessarily sharing similar interests.

5.6 Experimental Results for SVGIC-ST

In the following, we investigate the behavior of all methods in SVGIC-ST, where the default discount factor is 0.5, and the subgroup size constraint M is varied. As none of the baseline algorithms considers the subgroup size constraint, for SVGIC-ST with subgroup size constraint M, we first pre-partition the user set in to $\lceil \frac{N}{M} \rceil$ subgroups with balanced sizes. The results of all baseline algorithms on the partitioned subproblems are then aggregated into the final returned SAVG k-Configuration for them. Figures 9(a) and 9(b) show the total violation of subgroup size constraint (in total number of users) of all methods aggregated over all display slots in a total of 10 sampled instances in Timik (with $n = 25$) and Epinions (with $n = 15$), respectively. The suffix "-P" in the method name represents prepartitioning of the user set into $\lceil \frac{N}{M} \rceil$ subgroups with balanced sizes, and the suffix "-NP" indicates no prepartitioning is applied. The results manifest that the pre-partitioning helps decrease the total violation of the subgroup size constraint for all baseline methods except PER because the methods do not take the cap on subgroup sizes into account. AVG never violates the constraint since the greedy cutoff in CSF prevents large targeted subgroups. PER also achieves 100% feasibility since it does not consider social interactions. Among the baseline methods that consider social interactions, GRF incurs the lowest violation since only it partitions the user set into subgroups based on preference. However, it still has a low feasibility if not used with the pre-partition technique.

Figures 9(c) and 9(d) compares the total SAVG utility achieved by all methods *with pre-partitioning* in Timik and Epinions, respectively, with $n = 15$ and the subgroup size constraint varies from 3 to 15, where infeasible solutions achieves a total SAVG utility of 0. Note that all the baseline methods, except for PER, could still violate the subgroup size constraint even incorporated with the prepartitioning technique. This is because they from time to time display the same item to different pre-partitioned subgroups (or, in the case of GRF, subgroups of them) at the same display slot. AVG consistently outperforms all other methods except for the cases where M is very small (3 in Epinions). GRF achieves high total SAVG utility in Epinions as it achieves 100% feasibility and also selects proper distinct and preferred items for separated small subgroups of users in the sparse Epinions network. However, GRF has a surprisingly low feasiblity in Timik (around 20%), leading to a low total SAVG utility. This is because GRF displays the commonly-preferred popular VR locations in Timik to almost all pre-partitioned subgroups at the first few display slots due to its greedy (and thus deterministic) algorithmic behavior in selecting display items for subgroups. Therefore, while its accumulative violation is the lowest (among all methods that possibly violate the constraint), it often finds slightly oversized subgroups at slots 1 and 2, and thereby has a low feasibility.

5.7 User Study

For the user study, we recruit 44 participants[12] to visit our VR store. Their social networks, preference utility, and λ are pre-collected with questionnaires, which follows the setting of [42], and the social utility is learned by PIERT [31].[13] Finally, they are asked to provide λ in [0,1]. We investigate the following research question: After experiencing the VR store, are the participants satisfied with the SAVG k-Configurations generated by AVG, PER, FMG, and GRF? User feedbacks are collected in Likert score [52] from 1 to 5 (very unsatisfactory, unsatisfactory, average, satisfactory, and very satisfactory). Each group of participants visits the VR stores twice via hTC VIVE with the items selected by each scheme in randomized order.

Figure 10(a) reports that λ values specified by the users range from 0.15 to 0.85 with the average as 0.53, indicating that both personal preferences and social interactions are essential in VR group shopping. Figure 10(b) compares the total SAVG utility as well as the recorded user satisfaction

[12] The number of user participants is close to the user studies in VR research (e.g., 16 in [50]) and conventional group recommendation (e.g., 10 in [44] and 50 in [42]).

[13] A Likert scale questionnaire [52] is used to find the preference utility of items. Users are allowed to discuss about the products while filling the preference questionnaire so that the social utility can be learned from the products that are above average in terms of preference utility of a pair of friends [31].

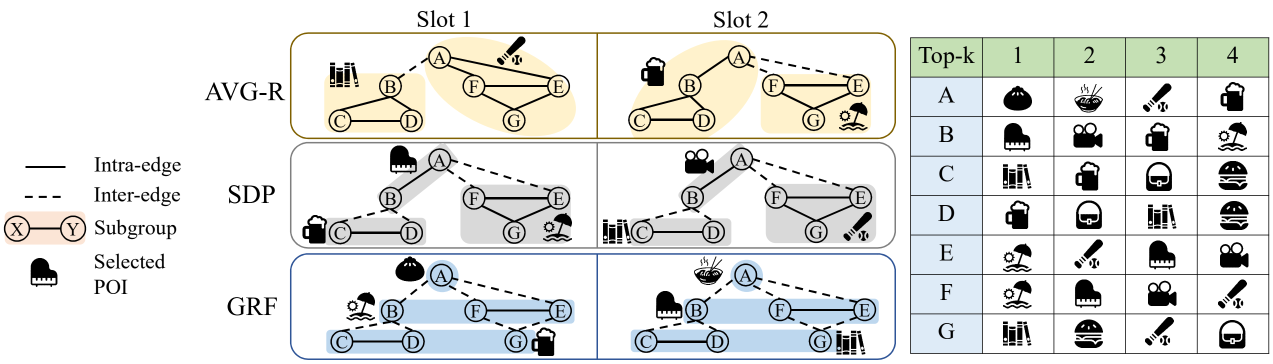
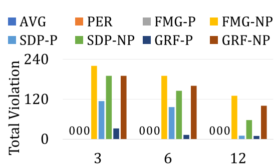
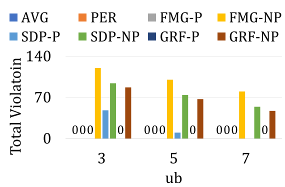


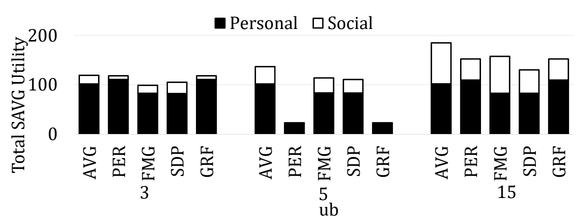
Figure 8: Illustration of subgroup partitioning approaches.



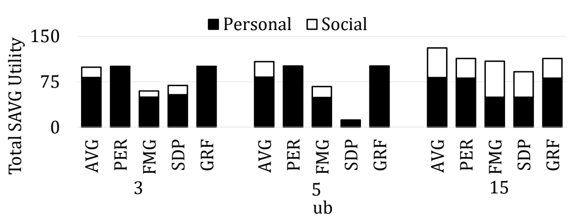
(a) Total violation vs. subgroup size constraint (Timik, $n = 25$).



(b) Total violation vs. subgroup size constraint (Epinions, $n = 15$).

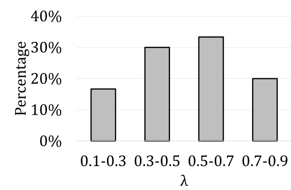


(c) Total SAVG utility vs. subgroup size constraint (Timik, $n = 15$).

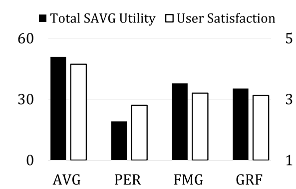


(d) Total SAVG utility vs. subgroup size constraint (Epinions, $n = 15$).

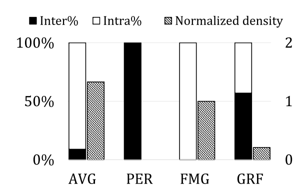
Figure 9: Comparisons on SVGIC-ST.



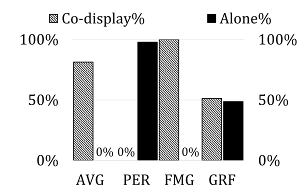
(a) Percentage of diff. λ.



(b) Total SAVG utility and user satisfaction.



(c) Inter/Intra% and subgroup density.



(d) Co-display rate and alone rate.

Figure 10: Comparisons on user study.

of each method. AVG outperforms the baselines by at least 34.2% and 29.6% in terms of the average total SAVG utility and average user satisfaction, respectively. The difference of AVG is statistically significant (p-value $\leq 0.019 < 0.05$). It is worth noting that we also calculate the correlation between the SAVG utility and user satisfaction (Spearman correlation 0.835; Pearson correlation 0.814), which manifest that the SAVG utility is a good estimation of user satisfaction.

Figures 10(c) and 10(d) report the subgroup metrics in the user study datasets. GRF, which separates users into subgroups according to preference similarities, returns a low normalized density (0.21), i.e., users in the same subgroup tends to be strangers. Compared with the results in large-scale datasets (Figure 7), GRF performs worse here since the normalized density is more sensitive when the user set is relatively small. In contrast, AVG flexibly assigns proper items to different subgroups of friends such that the normalized density is greater than 1 and the alone rate is 0%.

6. CONCLUSION

To the best of our knowledge, there exists no prior work tackling flexible configurations under the envisaged scenario of VR group shopping. In this paper, we formulate the SVGIC problem to retrieve the optimal SAVG k-Configuration that jointly maximizes the preference and the social utility, and prove its APX-hardness. We introduce an IP model and design a novel 4-approximation algorithm, AVG, and its deterministic version, AVG-D, by exploring the idea of Co-display Subgroup Formation (CSF) that forms subgroups of friends to display them the same items. Experimental results on real VR datasets manifest that our algorithms outperform baseline approaches by at least 30.1% in terms of solution quality.

7. REFERENCES

[1] Epinions dataset. http://www.trustlet.org/extended_epinions.html.

[2] Gurobi optimizer. http://www.gurobi.com. Accessed: 2019-09-30.

[3] IBM CPLEX optimizer. https://www.ibm.com/analytics/cplex-optimizer. Accessed: 2019-09-30.

[4] IrisVR. https://irisvr.com/. Accessed: 2019-09-30.

[5] VRChat. https://vrchat.net/. Accessed: 2019-09-30.

[6] Yelp dataset. https://www.yelp.com/dataset. Accessed: 2019-09-30.

[7] A. An, M. Kargar, and M. Zihayat. Finding affordable and collaborative teams from a network of experts. In *SDM*, pages 587–595. SIAM, 2013.

[8] E. Bozgeyikli, A. Raij, S. Katkoori, and R. V. Dubey. Point & teleport locomotion technique for virtual reality. In *CHI PLAY*, pages 205–216. ACM, 2016.

[9] C. Cai, R. He, and J. J. McAuley. SPMC: socially-aware personalized markov chains for sparse sequential recommendation. In *IJCAI*, pages 1476–1482, 2017.

[10] D. Cao, X. He, L. Miao, Y. An, C. Yang, and R. Hong. Attentive group recommendation. In *SIGIR*, pages 645–654. ACM, 2018.

[11] F. Chataigner, G. Manic, Y. Wakabayashi, and R. Yuster. Approximation algorithms and hardness results for the clique packing problem. *Discrete Applied Mathematics*, 157(7):1396–1406, 2009.

[12] H. Chen, D. Niu, K. Lai, Y. Xu, and M. Ardakani. Separating-plane factorization models: Scalable recommendation from one-class implicit feedback. In *CIKM*, pages 669–678. ACM, 2016.

[13] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, and T. Chua. Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention. In *SIGIR*, pages 335–344. ACM, 2017.

[14] L. Chen, C. Liu, R. Zhou, J. Li, X. Yang, and B. Wang. Maximum co-located community search in large scale social networks. *PVLDB*, 11(10):1233–1246, 2018.

[15] CNN. Alibaba offers VR shopping. https://edition.cnn.com/videos/world/2016/11/28/alibaba-vr-shopping-stevens-pkg.cnn. Accessed: 2019-09-30.

[16] M. B. Cohen, Y. T. Lee, and Z. Song. Solving linear programs in the current matrix multiplication time. In *IEEE FOCS*, 2019.

[17] eBay Australia. The world's first virtual reality department store - brought to you by eBay and myer. https://youtu.be/yAuiXhJPnr8. Accessed: 2019-09-30.

[18] S. Gidley. Virtual reality: Tourism firms use VR to attract visitors. https://bbc.in/2MExzZx. Accessed: 2019-09-30.

[19] R. Guerraoui, A. Kermarrec, T. Lin, and R. Patra. Heterogeneous recommendations: What you might like to read after watching interstellar. *PVLDB*, 10(10):1070–1081, 2017.

[20] A. Gulati and M. Eirinaki. With a little help from my friends (and their friends): Influence neighborhoods for social recommendations. In *WWW*, pages 2778–2784. ACM, 2019.

[21] R. He and J. J. McAuley. VBPR: visual bayesian personalized ranking from implicit feedback. In *AAAI*, pages 144–150. AAAI, 2016.

[22] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua. Neural collaborative filtering. In *WWW*, pages 173–182. ACM, 2017.

[23] L. Homaeian, N. Goyal, J. R. Wallace, and S. D. Scott. Group vs individual: Impact of TOUCH and TILT cross-device interactions on mixed-focus collaboration. In *CHI*, page 73. ACM, 2018.

[24] L. Hu, J. Cao, G. Xu, L. Cao, Z. Gu, and W. Cao. Deep modeling of group preferences for group-based recommendation. In *AAAI*, pages 1861–1867. AAAI, 2014.

[25] J. Jankowski, R. Michalski, and P. Bródka. A multilayer network dataset of interaction and influence spreading in a virtual world. *Scientific Data*, 4:170144, 2017.

[26] J. M. Kleinberg and É. Tardos. Approximation algorithms for classification problems with pairwise relationships: metric labeling and Markov random fields. *Journal of the ACM*, 49(5):616–639, 2002.

[27] S.-H. Ko, H.-C. Lai, H.-H. Shuai, D.-N. Yang, W.-C. Lee, and P. S. Yu. Optimizing item and subgroup configurations for VR shopping (full version). https://github.com/shaohengko/VRpaper, 2019.

[28] H. Li, Y. Ge, R. Hong, and H. Zhu. Point-of-interest recommendations: Learning potential check-ins from friends. In *KDD*, pages 975–984. ACM, 2016.

[29] J. Li, X. Wang, K. Deng, X. Yang, T. Sellis, and J. X. Yu. Most influential community search over large social networks. In *ICDE*, pages 871–882. IEEE, 2017.

[30] K. Li, W. Lu, S. Bhagat, L. V. S. Lakshmanan, and C. Yu. On social event organization. In *KDD*, pages 1206–1215. ACM, 2014.

[31] Y. Liao, W. Lam, L. Bing, and X. Shen. Joint modeling of participant influence and latent topics for recommendation in event-based social networks. *ACM Transactions on Information Systems*, 36(3):29:1–29:31, 2018.

[32] R. Lissermann, J. Huber, M. Schmitz, J. Steimle, and M. Mühlhäuser. Permulin: mixed-focus collaboration on multi-view tabletops. In *CHI*, pages 3191–3200. ACM, 2014.

[33] B. Lu, W. Fan, and M. Zhou. Social presence, trust, and social commerce purchase intention: An empirical research. *Computers in Human Behavior*, 56:225–237, 2016.

[34] Z. Lu, Z. Dou, J. Lian, X. Xie, and Q. Yang. Content-based collaborative filtering for news topic recommendation. In *AAAI*, pages 217–223. AAAI, 2015.

[35] X. Luo. How does shopping with others influence impulsive purchasing? *Journal of Consumer Psychology*, 15(4):288–294, 2005.

[36] P. Manurangsi. Almost-polynomial ratio eth-hardness of approximating densest k-subgraph. In *STOC*, pages 954–961. ACM, 2017.

[37] A. Nazi, Z. Zhou, S. Thirumuruganathan, N. Zhang,

and G. Das. Walk, not wait: Faster sampling over online social networks. *PVLDB*, 8(6):678–689, 2015.

[38] S. Qi, N. Mamoulis, E. Pitoura, and P. Tsaparas. Recommending packages to groups. In *ICDM*, pages 449–458. IEEE, 2016.

[39] S. S. Rangapuram, T. Bühler, and M. Hein. Towards realistic team formation in social networks based on densest subgraphs. In *WWW*, pages 1077–1088. ACM, 2013.

[40] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461. AUAI, 2009.

[41] Road to VR. Facebook social VR demo - oculus connect 2016. `https://youtu.be/YuIgyKLPt3s`. Accessed: 2019-09-30.

[42] S. B. Roy, L. V. S. Lakshmanan, and R. Liu. From group recommendations to group formation. In *SIGMOD*, pages 1603–1616. ACM, 2015.

[43] L. Q. Sánchez, J. A. Recio-García, B. Díaz-Agudo, and G. Jiménez-Díaz. Social factors in group recommender systems. *ACM Transactions on Intelligent Systems and Technology*, 4(1):8:1–8:30, 2013.

[44] D. Serbos, S. Qi, N. Mamoulis, E. Pitoura, and P. Tsaparas. Fairness in package-to-group recommendations. In *WWW*, pages 371–379. ACM, 2017.

[45] C. Shen, C. P. K. Fotsing, D. Yang, Y. Chen, and W. Lee. On organizing online soirees with live multi-streaming. In *AAAI*, pages 151–159. AAAI, 2018.

[46] C. Shen, D. Yang, L. Huang, W. Lee, and M. Chen. Socio-spatial group queries for impromptu activity planning. *IEEE Transactions on Knowledge and Data Engineering*, 28(1):196–210, 2016.

[47] C. Shen, D. Yang, W. Lee, and M. Chen. Spatial-proximity optimization for rapid task group deployment. *ACM Transactions on Knowledge Discovery from Data*, 10(4):47:1–47:36, 2016.

[48] H. Shuai, Y. Li, C. Feng, and W. Peng. Four-dimensional shopping mall: Sequential group willingness optimization under VR environments. In *WWW*, pages 131–134. ACM, 2018.

[49] H. Shuai, D. Yang, P. S. Yu, and M. Chen. Willingness optimization for social group activity. *PVLDB*, 7(4):253–264, 2013.

[50] V. Sitzmann et al. Saliency in VR: how do people explore virtual environments? *IEEE Transactions on Visualization and Computer Graphics*, 2018.

[51] TAKELEAP. Virtual reality store IKEA. `https://youtu.be/5_znFPj5Lis`. Accessed: 2019-09-30.

[52] W. M. Trochim and J. P. Donnelly. *Research methods knowledge base*, volume 2. Atomic Dog Publishing Cincinnati, OH, 2001.

[53] VentureBeat. Venturebeat: Amazon virtual reality VR mall kiosks - the future of retail. `https://youtu.be/J5NviNVdOsc`. Accessed: 2019-09-30.

[54] X. Wang, W. Zhu, C. Chen, and M. Ester. Joint user- and event- driven stable social event organization. In *WWW*, pages 1513–1522. ACM, 2018.

[55] D. Yang, C. Shen, W. Lee, and M. Chen. On socio-spatial group query for location-based social networks. In *KDD*, pages 949–957. ACM, 2012.

[56] M. Y.-C. Yim, S.-C. Yoo, P. L. Sauer, and J. H. Seo. Hedonic shopping motivation and co-shopper influence on utilitarian grocery shopping in superstores. *Journal of the Academy of Marketing Science*, 42(5):528–544, 2014.

[57] X. Zhang, S. Li, and R. R. Burke. Modeling the effects of dynamic group influence on shopper zone choice, purchase conversion, and spending. *Journal of the Academy of Marketing Science*, 46(6):1089–1107, 2018.

[58] X. Zhang, S. Li, R. R. Burke, and A. Leykin. An examination of social influence on shopper behavior using video tracking data. *Journal of Marketing*, 78(5):24–41, 2014.

[59] T. Zhao, J. J. McAuley, and I. King. Leveraging social connections to improve personalized ranking for collaborative filtering. In *CIKM*, pages 261–270. ACM, 2014.

[60] W. X. Zhao, Y. Guo, Y. He, H. Jiang, Y. Wu, and X. Li. We know what you want to buy: a demographic-based system for product recommendation on microblogs. In *KDD*, pages 1935–1944. ACM, 2014.

[61] W. X. Zhao, S. Li, Y. He, E. Y. Chang, J. Wen, and X. Li. Connecting social media to e-commerce: Cold-start product recommendation using microblogging information. *IEEE Transactions on Knowledge and Data Engineering*, 28(5):1147–1159, 2016.