

## Response to Reviewer Comments

**Reviewer 1, detailed 1 (R1D1):** The motivation is still weak. First, the target scenario is quite specific. It seems that the proposed problem can only be used for VR group shopping. Second, even in the VR shopping scenario, the problem depends on the customized interactive display and social interactions. Authors need to justify that the target application is a big trend. Whether the proposed model and solution can be used in other application also needs to be sufficiently discussed.

**Reply:** Thanks for the valuable comments. As suggested, we have 1) elaborated more on the growing trend of *VR shopping*, *customized display*, and *social interactions*, and 2) identified *Social Event Organization* (SEO) as another potential application of the proposed problem.

*VR Shopping.* A marketing report about future retails from Oracle [8] manifests that 78% of online retailers already have implemented or are planning to implement VR and AI. Recently, International Data Corporation (IDC) forecasts the worldwide spending on VR/AR to reach 18.8 billion USD in 2020 [87], including \$1.5 billion in retails. It also foresees the VR/AR market to continue an annual growth rate of 77% through at least 2023. Also, Goldman Sachs forecasts the global market for VR/AR in retails to reach USD 1.6 billion by 2025 [64].

*Customized Display.* IKEA (see video [81] at 0:40) and Lowe's [49] respectively launch VR store applications where the displayed furniture may adapt to the user preferences, which attracts more than 73% of the users to recommend IKEA VR store [38]. Alibaba's [24] and eBay's VR department stores [53] also devote themselves to provide personalized product offers. According to a marketing survey, 79% of surveyed US, UK, and China consumers are likely to visit a VR store displaying customized products [25].

*Social Interactions.* A survey of 1,400 consumers in 2017 shows that 65% of the consumers are excited about VR shopping, while 54% of them acknowledge social shopping as their ways to purchase products [58]. The L.E.K. consulting survey [34] shows that 70% of 1,000 consumers who had already experienced VR technology are strongly interested in virtual shopping with friends who are not physically present. Embracing the trend, Shopify [74] and its technical partner Qbit [59] build a social VR store supporting attendance of multiple users with customized display [1].

Encouraged by the above evidence, we make the first attempt to formulate the problem for group shopping in VR stores. Note that the SVGIC problem supports simpler item recommendation tasks as special cases, i.e., shopping in brick-and-mortar stores, conventional e-commerce personal item recommendation, and group recommendation in VR without CID can be modeled as special cases of SVGIC. As suggested, the above clarification has been included in Section 3.1 in the paper.

As suggested, we have also identified *Social Event Organization* (SEO) as another important application of the targeted problem. Specifically, SEO [14, 46, 69, 70, 85] studies the problem of organizing overlapping and conflicting events for users of Event-Based Social Network (EBSN) (e.g. Meetup [56], Facebook Events [21], and Douban Location [17]). Given a set of users and a set of events, SEO assigns each user to a series of events, such that the total preferences of the selected users to their assigned events are

maximized, while several constraints are satisfied, e.g. traveling cost budget, time conflict avoidance, and event sizes. Consequently, the optimization problems for SEO are closely related to SVGIC: social event attendees directly correspond to VR shopping users, while social events correspond to the displayed items in VR. SVGIC can be viewed as a general version of the SEO problem that incorporates social-based utilities to support a series of social events. As suggested, the above explanations have been added to Section 4.4 of the revised paper. Moreover, in Section 4.4 of the full version [42], we explicitly discuss how to adapt our proposed AVG algorithm for an advanced problem, SVGIC-ST, that supports an additional group size constraint that suits for SEO.

**Reviewer 1, detailed 2 (R1D2):** In general, the problem model is not well studied from the application perspective. The complexity perspective of the proposed problem model and corresponding solution is well studied. However, why the constraints/assumptions/objectives are reasonable in real applications is not sufficiently discussed.

**Reply:** Thanks for the valuable comments. The assumptions and problem formulation are explained as follows.

1) *The CID-supportive VR Group Shopping Scenario.* Marketing survey manifests that 70% of VR users are strongly interested in interactive VR group shopping [34]; thus we envision the upcoming of CID-supportive VR shopping systems. Specifically, the VR technology supporting social interactions has already been employed in existing VR social platforms, e.g., VRChat [83], AltspaceVR [3], and Facebook Horizon [22], where users, represented by customized avatars, interact with friends in various ways: 1) *Movement and gesture.* Commercial VR devices, e.g., Oculus Quest and HTC Vive Pro/Focus, support six-degree-of-freedom (6DOF) headsets and hand-held controllers to detect the user movements and gestures in real-time. These movements are immediately updated to the server and then perceived by other users. For example, a high-five gesture mechanism is used in Rec Room [62] to let users befriend others. 2) *Emotes and Emojis.* Similar to emoticons in social network platforms, VR users can present their emotions through basic expressions (happy face, sad face, etc.) or advanced motions (e.g., dance, backflip, and die in VR-Chat [84]). 3) *Real sound.* This allows users to communicate with friends verbally through their microphones. As such, current VR applications support immersive and seamless social interactions, rendering social interactions realistic in shopping systems.

Moreover, under the scenario of brick-and-mortar group shopping, previous research [50, 52, 66, 91, 94] indicates that the satisfaction of a group shopping user is affected by two factors: *personal preference* and *social interaction*. Furthermore, empirical research on dynamic group interaction in retails finds that shopping while discussing with others consistently boosts sales [91], while intra-group talk frequency has a significant impact on users' shopping tendencies and purchase frequencies [94, 95].

2) *Objective Functions.* According to previous research [50, 52, 66, 91, 94], we capture the overall satisfaction by combining 1) the aggregated personal preferences and 2) the retailing benefits from facilitating social interactions and discussions. We note that, from an optimization perspective, these two goals form a trade-off because close

friends/families may have diverse personal preferences over items. One possible way to simultaneously consider both factors is to use an end-to-end machine learning-based approach, which generates the *user embeddings* and *item embeddings* [9] and designs a neural network aggregator to derive the total user satisfaction. However, this approach requires an algorithm to generate candidate configurations for ranking and relies on a huge amount of data to train the parameters in the aggregator. On the other hand, previous research [76, 85] has demonstrated that a *weighted combination* of the preferential and social factors is effective in measuring user satisfaction, where the weights can be directly set by a user or implicitly learned from existing models [47, 97]. Another possible objective is a weighted sum of the previous objective metric (the aggregated preference) with the aggregated social satisfaction as the constraint, or vice versa, which is equivalent to our problem after applying Lagrangian relaxation to the constraint. Therefore, we follow [76, 85] to define the SAVG utility as a combination of *aggregated preference utility* and *aggregated social utility*, weighted by a parameter  $\lambda$ . To validate the effectiveness of this objective model, we have conducted a user study in a real CID VR shopping system, which shows high correlations between user satisfaction and the proposed objective (Please see the response for R1D10 and Section 6.9 in the revised paper for details).

As suggested, the above clarification has been added to Section 3.1; for the discussion on the notions of *direct co-display* and *indirect co-display*, please see the response for R1D3.

**Reviewer 1, detailed 3 (R1D3): Most constraints in the modeling process increased difficulty of the problem, e.g. the common item for a pair of friends should be in same slot. It is required to show their necessity in modelling a real-world problem.**

**Reply:** Thanks for the valuable comments. We have provided the connection between the co-display notion and the practical scenario as follows, whereas the clarification of other constraints are presented in the response to R1D2 above and included in Section 3 of the revised paper.

With non-customized user environments in brick-and-mortar stores, discussing a specific item near its location is the default behavior. On the other hand, as users are displayed completely customized items in personalized e-commerce (e.g., the 20 items appearing on the main page of a shopping site can be completely disjoint for two friends), it often requires a “calibration” step (e.g., sharing the exact hyperlink of the product page) to share the view on the target item before e-commerce users can engage in discussions on some items. Since shopping users are used to having discussions on the common item they are viewing at the identical slot in their own Virtual Environments (VEs), we devote to enhance the possibility of this intuitive *direct* setting in our envisioned VR shopping VEs even though other settings are also possible in VR.

Therefore, in the general setting where each user is displayed  $k$  different items at  $k$  display slots, for a pair of friends displayed a common item, it is most convenient to *directly* display the item at the same slot in the two users’ respective VEs such that they can locate the exact position easily (e.g., “come to the second shelf; you should see this.”). Accordingly, upon viewing any item, it is expected that the user interface indicates a list of friends being co-displayed

the specific item with the current user, so that she is aware of the potential candidates to start a discussion with.

In contrast, if a common item is displayed at different slots in two users’ VEs, a prompt discussion is more difficult to initiate since the users need to locate and move to the exact position of the item first. Nonetheless, to model this event that requires more efforts for users, we have enriched the problem by introducing the notion of *indirect co-display* into the more generalized SVGIC-ST problem, where items displayed at different slots generate a slightly discounted social utility (i.e., contribute less than those *directly* co-displayed to users) due to the inconvenience from the need of calibrating item locations through VR functions (e.g, teleportation) before the discussion.

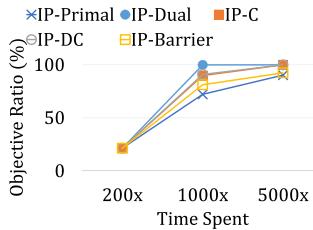
Finally, although the difference between direct and indirect co-display does not seem to be much in the illustrative example with  $k = 3$  slots, the aforementioned issue becomes severe for a large-scale VR shopping mall (say,  $k = 50$ ). As suggested, the above clarification has been added to Section 3.1 of the revised paper, and we have also extended AVG for SVGIC-ST in Section 4.4.

**Reviewer 1, detailed 4 (R1D4): Since the problem is modeled as integer programming, there are some approximate algorithms for integer programming with faster speed and lower results. They should be added as competitors.**

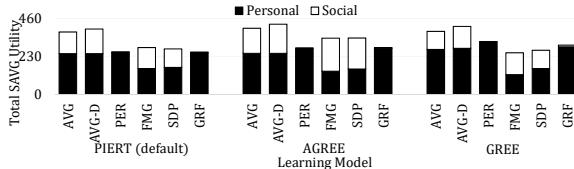
**Reply:** Thanks for the valuable comments. We have accordingly conducted experiments with approximate algorithms for mixed integer programming (MIP) algorithms in the Gurobi [29] package. The algorithms used are Primal-first (IP-Primal), Dual-first (IP-Dual), Concurrent (IP-C), Deterministic Concurrent (IP-DC), and the Barrier Algorithm (IP-Barrier). Figure R.1 shows the trade-off between efficiency and efficacy of five different IP algorithms on the Timik dataset with the default parameters ( $(k, m, n) = (50, 10000, 125)$ ). For every MIP algorithm, we evaluate the solution quality of different algorithms with the running time constrained by 200, 1000 and 5000 times the running time of our proposed AVG-D algorithm for the same instance to compare the obtained solutions in different time limits. The y-axis shows the objective value normalized by the solution of AVG-D, which manifests none of the 5 baselines achieves any solution better than that of AVG-D in 5000X of the running time of AVG-D. As such, although there is some subtle difference in performance across different MIP algorithms, none of the examined algorithms shows a reasonable scalability. As suggested, the above new result has been added to Section 6.4 of the revised paper.

**Reviewer 1, detailed 5 (R1D5): The  $\tau$  matrix (social utility) is too large. Is it easy to be retrieved in real-world situation? Is it necessary to define different weights for each friendship-item pair?**

**Reply:** We agree that the  $\tau$  matrix is often large in large-scale inputs. Indeed, there exist simpler alternatives for modeling the social utility in large-scale applications. For instance, one possible way is to regard the social utility  $\tau(c, u, v)$  as the product of user  $u$ ’s preference utility  $p(u, c)$  and the *social trust*  $\theta(u, v)$  from user  $u$  to  $v$ , i.e., the degree of how a user trusts another on recommended items [28, 89]. In this alternative formulation, the social utility can be derived from an  $n \times n$ -matrix (specifying  $\theta(u, v)$ ) and the preferences, instead of from an  $n \times n \times m$ -tensor.



**Figure R.1: Results of different IP heuristics.**



**Figure R.2: Total SAVG utility vs. different input.**

According to these alternatives, it might not be necessary to define different weights for each friendship-item pair. However, we define the  $\tau$  matrix as the most general form of input and design AVG and AVG-D to handle the most general case. Note that SVGIC admits the same combinatorial hardness and algorithmic result even with this variant of input.

**Reviewer 1, detailed 6 (R1D6): The parameters setting of  $\tau$  and  $p$  in the datasets should be given.**

**Reply:** Thanks for the comment. We have conducted experiments on different input models. Please see the response for R1D7.

**Reviewer 1, detailed 7 (R1D7): The effectiveness of the proposed model/solution heavily depends on the input social utility and user preference, while these two inputs are generated by other literatures. The influence of the input should be well addressed.**

**Reply:** Thanks for the valuable comments. Accordingly, we have conducted experiments on different inputs generated by PIERT [47] (default), AGREE and GREE [9]. PIERT jointly learns the preference and social utilities by modeling the social influence between users and the latent topics of items. For AGREE and GREE [9], the former uses an attention network to learn sophisticated social utilities of the triple (user, user, item), and the latter assumes a uniform social influence between any pair of users. As shown in Figure R.2, AVG and AVG-D outperform the baselines with regards to all different input models. The social utilities returned by AVG and AVG-D with AGREE are slightly greater than the ones with GREE due to the sophisticated social utilities in AGREE. As suggested, the above new result has been incorporated in Section 6.3 of the revised paper.

**Reviewer 1, detailed 8 (R1D8): Some minor errors: e.g. p11, “Figures 5 compares the results on”.**

**Reply:** Thanks for pointing out the typo. We have fixed them.

**Reviewer 1, detailed 9 (R1D9): The organization of the paper could be further improved. The introduction part is not easy to follow.**

**Reply:** Thanks for the valuable comments. As suggested, we have re-organized the paper to include more elabora-

tions and discussions on the application, especially in the introduction and problem formulation sections, while moving some technical details to the full version of the paper [42]. Moreover, we have enriched the motivating example (Figure 1 in Section 1) for clearer illustration.

**Reviewer 1, detailed 10 (R1D10): The evaluation metrics of preference utility and social utility are exactly the objectives of the model. It is not surprising to see that the model beats other baselines for these two metrics. Are there any other ways to improve these experiments?**

**Reply:** Thank you! As suggested, we have built a prototype of VR store with Unity 2017.1.1.1, SteamVR 1.2.2, and 3ds Max 2016 for HTC VIVE HMD, and conducted real user studies to measure the user satisfaction in viewing SAVG  $k$ -Configurations generated by AVG and three baselines PER, FMG, and GRF. A Likert questionnaire [79] is used to find the preference utility of items. Users are allowed to discuss the products so that the social utility can be learned by PIERT [47]. Experimental results manifest that the recorded user satisfaction of AVG outperforms the baselines by at least 29.6%, while the difference is statistically significant ( $p$ -value  $\leq 0.019 < 0.05$ ). The correlation coefficients between the SAVG utility and user satisfaction are respectively 0.835 and 0.814 for Spearman and Pearson correlations, which validate our proposed utility model. Please see Section 6.9 in the revised paper for the details of the user study.

In addition to the objective, as suggested, we have measured a set of group formation-related performance metrics, including 1) the percentage of *Inter*-subgroup edges and *Intra*-subgroup edges in the solutions, 2) the average network density among partitioned subgroups, normalized by the average density of the original social network, 3) the percentage of friend pairs viewing common items together (*Co-display%*), 4) the percentage of users viewing items alone (*Alone%*), and 5) *regret ratio*, a game-theoretic measurement for satisfaction of individual users and the overall *fairness* of the solution. To summarize, AVG achieves the largest normalized density, a high *Co-display%*, and a near-zero *Alone%*. The regret ratios in both AVG and AVG-D seldom exceed 20%, while some baselines have regrets much larger than 20%. Please see Section 6.5 in the revised paper for complete experimental results.

**Reviewer 1, detailed 11 (R1D11): The configuration is a typical combinatorial optimization problem. The related work should cover literatures in this direction. The contribution should also be clarified from this perspective.**

**Reply:** Thanks for the valuable comments. Indeed, SVGIC is related to the *Multi-Labeling* (ML) [88] problem and its variations, including Multiway Partition [88, 96], *Maximum Happy Vertices/Edges* (MHV/MHE) [92], and *Multiway Cut* [16] in graphs.

In the *Multi-Labeling* (ML) problem, the general inputs are a ground set of entities  $V$ , a set of labels  $L = \{1, 2, \dots, m\}$ , and a *partial labeling* function  $\ell : V \mapsto L$  that pre-assigns each label  $i$  to a non-empty subset  $T_i \subseteq V$ , and a set function  $f : 2^V \mapsto \mathcal{R}$ . The goal of ML is to partition the ground set into  $m$  subsets to optimize the aggregated set function on the partitioned subsets, i.e.,  $\sum_{i=1}^m f(T_i)$ . The

**Table R.1: Related Combinatorial Problems (approximation ratio for MAX problems inversed).**

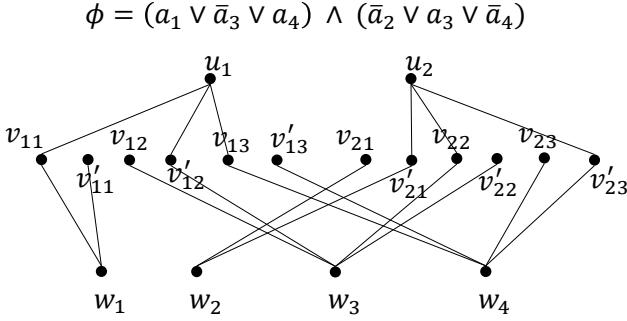
Problem Name	Abbrev.	Prelabeling	Type	Objective	Hardness Result	Approx. Result	Is a special case of	Is the complement of
Submodular Multi-Labeling [88]	Sub-ML	General	MIN	submodular set function	$2 - \frac{2}{m} - \epsilon$ (UGC) [88]	$2 - \frac{2}{m}$ [88]		Sup-ML
Supermodular Multi-Labeling [88]	Sup-ML	General	MAX	supermodular set function	NP-hard	$\frac{m}{2}$ [88]		Sub-ML
Submodular Multiway Partition [96]	Sub-MP	$m$ terminals	MIN	submodular set function	$2 - \frac{2}{m} - \epsilon$ (NP = RP) [19]	$2 - \frac{2}{m}$ [19]	Sub-ML	Sup-MP
Supermodular Multiway Partition [88]	Sup-MP	$m$ terminals	MAX	supermodular set function	NP-hard		Sup-ML	Sub-MP
Maximum Happy Vertices [92]	MHV	General	MAX	# happy vertices	NP-hard	$\frac{m}{2}$ [88]	Sup-ML	MUHV
Minimum Unhappy Vertices [88]	MUHV	General	MIN	# unhappy vertices	$2 - \frac{2}{m} - \epsilon$ (UGC) [88]	$2 - \frac{2}{m}$ [88]	Sub-ML	MHV
Maximum Happy Edges [92]	MHE	General	MAX	# happy edges	NP-hard	1.1716 [93]	Sup-ML	
Multiway Cut [16]	$m$ terminals		MIN	# edges removed	$\frac{8}{7}$ [54]	1.2965 [68]	UML	Multiway Uncut
Multiway Uncut [43]	$m$ terminals		MAX	# edges preserved	NP-hard	1.1716 [43]	MHE	Multiway Cut
Uniform Metric Labeling [41]	UML	$m$ terminals	MIN	# total cost of edges/vertices	$2 - \frac{2}{m}$ (UGC) [54]	2 [41]	Metric Labeling	

special cases with set function  $f$  being submodular or supermodular is called *Submodular Multi-Labeling* (Sub-ML) or *Supermodular Multi-Labeling* (Sup-ML) [88]. Another work [96] studies the case where the partial labeling function is required to assign a label  $i$  to only one entity  $v_i$ . The above problems become *Submodular Multiway Partition* (Sub-MP) and *Supermodular Multiway Partition* (Sup-MP) in this context. Some special cases of Sub-ML and Sup-ML are described as graph problems, where the ground set  $V$  is the vertex set on a graph  $G = (V, E)$ . In this context, the above assumption on  $f$  in Sub-MP and Sup-MP can be interpreted as  $m$  specified terminals on the graph. Two representative special cases are the *Maximum Happy Vertices* (MHV) problem and the *Maximum Happy Edges* (MHE) problem [92]. Originally proposed to study homophily effects in networks, MHV and MHE are described as coloring problems where, different from traditional graph coloring, vertices are encouraged to share the same color with neighbors. *Edges with same-color vertices* and *vertices sharing the same color with all neighbors* are characterized as *happy edges* and *happy vertices*, respectively. The optimization goals of MHV and MHE are thus to maximize the number of happy vertices and edges. The complement problem of MHV, *Minimum Unhappy Vertices* (MUHV) [88], minimizes the number of unhappy vertices. When the above optimization goal of MHE is combined with the assumption of  $m$  terminals, the resultant problem becomes the *Multiway Uncut* problem [43], which is the complement of the classical and extensively studied *Multiway Cut* problem [16]. The Multiway Cut problem asks for the minimum number of edges removed to partition a graph into  $m$  subgraph, each containing one terminal. The Metric Labeling problem [41] is a generalization of Multiway Cut considering edge weights and also a cost function on labeling the vertices. We summarize the problem characteristics, hardness results, currently best algorithmic results, and their interdependency in Table R.1. All the problems are NP-hard, while more advanced inapproximability results, relying on other complexity con-

jectures, are stated for some problems. Note that the approximation ratios for maximization problems are shown as inverses (i.e., values larger than 1) for better comparison and consistency with this paper.

Among the ML-type problems, SVGIC is particularly correlated to the MHE problem. Regarding the displayed items in SVGIC as colors (labels) in MHE, the social utility achieved in SVGIC is closely related to the number of preserved happy edges in MHE, and SVGIC thereby encourages partitioning all users into dense subgroups to preserve the most social utility. However, SVGIC is more difficult than the above ML-type problems due to the following reasons. 1) The ML-type problems find a strict partition that maps each entity/vertex to only one color (label), while SVGIC assigns  $k$  items to each user, implying that any direct reduction from the above problems can only map to the  $k = 1$  special case in SVGIC. 2) The above problems do not discriminate between different labels, i.e., switching the assigned labels of two different subgroups does not change their objective functions. This corresponds to the special case of SVGIC where all preference utility  $p(u, c)$  and social utility  $\tau(u, v, c)$  do not depend on the item  $c$ . In contrast, SVGIC captures the item preferences as both utility measures in SVGIC depend on the item identity. 3) The above problems admit a partial labeling (pre-labeling) in the input such that some entities have predefined fixed labels (otherwise labeling every entity with the same label is optimal, rendering the problem trivial), while SVGIC does not specify any item to be displayed to specific users. However, SVGIC requires  $k$  items for each user; moreover, even in the  $k = 1$  special case, simply displaying the same item to all users is not optimal in SVGIC due to the item-dependent preference and social utility.

Finally, we note that while most of the minimization problems admit stronger hardness results than conventional NP-hardness, no such type of results are known for all maximization problems above. As SVGIC is also a maximization problem, its hardness does not directly follow from any complementary minimization problems.



**Figure R.3:** An illustration instance built for MAX – E3SAT

**Reviewer 2, weak 1 (R2W1):** The approximate ratio (i.e. 4) seems to be loose.

**Reply:** Thanks for the comment. As suggested, to enhance the theoretical results, we improve the hardness of approximation of SVGIC through a gap-preserving reduction from the MAX – E3SAT problem [30] and discuss the approximation ratio.

In the following, we prove the hardness result with a gap-preserving reduction from the MAX – E3SAT problem [30]. Given a conjunctive normal form (CNF) formula  $\phi$  with exactly 3 literals in each clause, MAX – E3SAT asks for a truth assignment of all variables such that the number of satisfied clauses in  $\phi$  is maximized. The following lemma (from [30]) states the hardness of approximation of MAX – E3SAT.

**Lemma R.1.** MAX – E3SAT is NP-hard to approximate within a ratio of  $\frac{8}{7} - \epsilon$ .

Our reduction builds on the following lemma.

**Lemma R.2.** There exists a gap-preserving reduction from MAX – E3SAT to SVGIC which translates from a formula  $\phi$  to an SVGIC instance  $\mathbf{I}$  such that

- If at least  $\chi$  clauses can be satisfied in  $\phi$ , then the optimal value of  $\mathbf{I}$  is at least  $8 \cdot \chi$ ,
- If less than  $\frac{7}{8} \cdot \chi$  clauses can be satisfied in  $\phi$ , then the optimal value of  $\mathbf{I}$  is less than  $\frac{31}{4} \cdot \chi$ .

*Proof.* Given a MAX – E3SAT instance  $\phi$  with  $n_{\text{var}}$  Boolean variables  $a_1, \dots, a_{n_{\text{var}}}$  and  $m_{\text{cla}}$  clauses  $C_1, \dots, C_{m_{\text{cla}}}$ , we construct an SVGIC instance as follows. 1) Each clause  $C_j = (l_{j1} \vee l_{j2} \vee l_{j3})$  corresponds to i) a vertex  $u_j \in V_1$ , and ii) six vertices  $v_{j1}, v'_{j1}, v_{j2}, v'_{j2}, v_{j3}, v'_{j3} \in V_2$ , where  $v_{j_t}$  corresponds to  $a_i$ , and  $v'_{j_t}$  corresponds to  $\bar{a}_i$  (the negation of  $a_i$ ), if and only if  $l_{jt}$  is  $a_i$  or  $\bar{a}_i$ . 2) Each Boolean variable  $a_i$  corresponds to a vertex  $w_i \in V_3$ . (Thus,  $V = (V_1 \cup V_2 \cup V_3)$  has  $(n_{\text{var}} + 7 \cdot m_{\text{cla}})$  vertices.) 3) For each clause  $C_j$ , we add an edge between  $u_j$  and each of the three vertices in  $V_2$  that correspond to TRUE assignments of the literals in  $C_j$ . 4) For each literal  $l_{jt}$ , if  $l_{jt}$  is  $a_i$  or  $\bar{a}_i$ , we add an edge between  $w_i$  and  $v_{jt}$ , and also an edge between  $w_i$  and  $v'_{jt}$ . (Therefore,  $G$  has  $(9 \cdot m_{\text{cla}})$  edges.) 5) For each edge between  $V_1$  and  $V_2$  with the form  $(u_j, v_{jt})$ , we add an item  $c_{jt}$ ; similarly, for each edge between  $V_1$  and  $V_2$  with the form  $(u_j, v'_{jt})$ , we add an item  $c'_{jt}$ . 6) For each Boolean variable  $a_i$ , we add two items  $c_i$  and  $c'_i$ . 7) For the preference utility values, let  $p(v, c) = 0$  for all  $v \in V, c \in \mathcal{C}$ . 8) For

the social utility values, let  $\tau(u_j, v_{jt}, c_{jt}) = \tau(v_{jt}, u_j, c_{jt}) = 1$  for every  $(u_j, v_{jt}) \in E$ ; similarly, let  $\tau(u_j, v'_{jt}, c'_{jt}) = \tau(v'_{jt}, u_j, c'_{jt}) = 1$  for every  $(u_j, v'_{jt}) \in E$ . Moreover, let  $\tau(w_i, v_{jt}, c_i) = \tau(v_{jt}, w_i, c_i) = 1$  if and only if  $l_{jt} = a_i$ ; similarly, let  $\tau(w_i, v'_{jt}, c'_i) = \tau(v'_{jt}, w_i, c'_i) = 1$  if and only if  $l_{jt} = \bar{a}_i$ . All other  $\tau$  values are zero. An example is illustrated in Figure R.3. Finally, let  $k = \lambda = 1$ .

We first prove the sufficient condition. Assume that there exists a truth assignment in  $\phi$  satisfying at least  $m$  clauses. We construct a feasible solution for SVGIC as follows. 1) For each satisfied clause  $C_j$ , let  $t_j$  be the smallest  $t$  such that  $l_{jt}$  is assigned TRUE.  $u_j$  is displayed item  $c_{jt}$  if  $l_{jt}$  is of the form  $a_i$ ; otherwise  $u_j$  is displayed item  $c'_{jt}$ ; 2) For each literal  $l_{jt}$  with the form  $a_i$  and assigned TRUE,  $v_{jt}$  is displayed item  $c_{jt}$ ; for each literal  $l_{jt}$  with the form  $\bar{a}_i$  and assigned TRUE,  $v'_{jt}$  is displayed item  $c'_{jt}$ ; 3) For each literal  $l_{jt}$  with the form  $a_i$  assigned FALSE,  $v_{jt}$  is displayed item  $c_i$ ; for each literal  $l_{jt}$  with the form  $\bar{a}_i$  assigned FALSE,  $v'_{jt}$  is displayed item  $c'_i$ ; 4) For each Boolean variable  $a_i$ ,  $w_i$  is displayed item  $c_i$  if and only if  $a_i$  is FALSE in the assignment; otherwise  $w_i$  is displayed item  $c'_i$ ; 5) For an unsatisfied clause  $C_j$ ,  $u_j$  can be displayed any item.

In the following, we prove that this solution achieves an objective of at least  $(2 \cdot m + 6 \cdot m_{\text{cla}})$  in SVGIC. We first observe that for each satisfied clause  $C_j$ , either  $c_{jt}$  is co-displayed to  $u_j$  and  $v_{jt}$ , or  $c'_{jt}$  is co-displayed to  $u_j$  and  $v'_{jt}$ ; each of the above cases achieves an objective value of 2. As there are at least  $m$  satisfied clauses, the above two cases in total contribute at least  $2 \cdot m$  to the objective of SVGIC. Next, for each Boolean variable  $a_i$  assigned TRUE,  $w_i$  is co-displayed  $c_i$  with every  $v'_{jt}$  such that  $l_{jt}$  is  $a_i$  or  $\bar{a}_i$ ; similarly, for each Boolean variable  $a_i$  assigned FALSE,  $w_i$  is co-displayed  $c_i$  with every  $v_{jt}$  such that  $l_{jt}$  is  $a_i$  or  $\bar{a}_i$ . The above two cases achieve an objective value of 2 for each co-display pairs. As there are exactly  $(3 \cdot m_{\text{cla}})$  such pairs of  $(w_i, v_{jt})$  or  $(w_i, v'_{jt})$ , they in total contributes  $(6 \cdot m_{\text{cla}})$  to the objective of SVGIC. Combining the above two parts, the objective value is at least  $(2 \cdot m + 6 \cdot m_{\text{cla}})$ .

We then prove the necessary condition. Assume that all possible truth assignments in  $\phi$  satisfy fewer than  $m'$  clauses. We then prove that the optimal solution in the SVGIC instance is strictly less than  $(2 \cdot m' + 6 \cdot m_{\text{cla}})$ . We prove it by contradiction. If there exists a feasible solution of SVGIC with an objective at least  $(2 \cdot m' + 6 \cdot m_{\text{cla}})$ , we prove that there exists at least one feasible solution of SVGIC such that the total social utility from co-displaying items to one vertex in  $V_2$  and another vertex in  $V_3$ , denoted by  $\tau(V_2, V_3)$ , is exactly  $6 \cdot m_{\text{cla}}$ . To prove the above argument, we first observe that the total number of edges between vertices in  $V_2$  and vertices in  $V_3$  is  $6 \cdot m_{\text{cla}}$ . Furthermore, these edges form exactly  $3 \cdot m_{\text{cla}}$  copies of  $P_3$  (where every  $P_3$  is formed by a  $w_i$  and two vertices  $v_{jt}$  and  $v'_{jt}$  such that  $l_{jt}$  is  $a_i$  or is  $\bar{a}_i$ ). Since the two edges in each  $P_3$  can only contribute to the final objective through different co-displayed items, these edges in total contribute at most  $6 \cdot m_{\text{cla}}$  to the final objective. Therefore,  $\tau(V_2, V_3) \leq 6 \cdot m_{\text{cla}}$ . Next, suppose a solution with  $\tau(V_2, V_3) < 6 \cdot m_{\text{cla}}$  is given. For an arbitrary literal  $l_{jt}$ , let  $a_i$  be the corresponding variable. We examine the following cases: 1)  $w_i$  is co-displayed  $c_i$  with  $v_{jt}$  or  $c'_i$  with  $v'_{jt}$ . In this case, a social utility of 2 is achieved. 2)  $w_i$  is not co-displayed any item with  $v_{jt}$  or  $v'_{jt}$ . As  $w_i$  does not contribute to the objective if it is displayed neither  $c_i$  nor  $c'_i$ ,

it is feasible to display  $c_i$  to  $w_i$  without decreasing the total objective. Next, it is also feasible to co-display  $c_i$  to *both*  $v_{j_t}$  and  $v'_{j_t}$  while not decreasing the total objective, since the most social utility that  $v_{j_t}$  or  $v'_{j_t}$  could contribute previously from being co-displayed  $c_{j_t}$  (or  $c'_{j_t}$ ) is also at most 2. By repeatedly applying the two cases, we conclude that there exists a solution where each literal  $1_{j_t}$  contributes exactly 2 to  $\tau(V_2, V_3)$ , which implies  $\tau(V_2, V_3) = 6 \cdot m_{\text{cla}}$ .

Given a feasible solution with  $\tau(V_2, V_3) = 6 \cdot m_{\text{cla}}$ , we construct a truth assignment in  $\phi$  according to this solution as follows. Let every variable  $a_i$  be **TRUE** if and only if  $w_i$  is displayed  $c'_i$ ; otherwise  $a_i$  is **FALSE**. Note that this is consistent to the construction in the sufficient condition. For each pair of vertices  $(u_j, v_{j_t})$  co-displayed  $c_{j_t}$  in the solution, since  $v_{j_t}$  is not co-displayed an item with any vertex in  $V_3$ , it follows that  $v'_{j_t}$  must be co-displayed  $c'_i$  with some  $w_i \in V_3$  (otherwise  $\tau(V_2, V_3) = 6 \cdot m_{\text{cla}}$  does not hold), which in turn implies  $a_i$  (appearing in clause  $C_j$ ) is assigned **TRUE** in this truth assignment, and  $C_j$  is therefore satisfied. Analogously, for  $(u_j, v'_{j_t})$  co-displayed  $c'_{j_t}$  in the solution,  $C_j$  is also satisfied by this truth assignment. As the total objective is at least  $(2 \cdot m' + 6 \cdot m_{\text{cla}})$  and  $\tau(V_2, V_3) = 6 \cdot m_{\text{cla}}$ , the total social utility from co-displaying items to one vertex in  $V_1$  and another vertex in  $V_2$ , denoted by  $\tau(V_1, V_2)$ , is at least  $(2 \cdot m')$ , meaning that at least  $m'$  clauses are satisfied in  $\phi$  by the constructed truth assignment, leading to a contradiction. Therefore, we conclude that the optimal solution in the SVGIC instance is strictly less than  $(2 \cdot m' + 6 \cdot m_{\text{cla}})$ .

Finally, for any instance of MAX – E3SAT, a folklore random assignment of variables satisfies  $\frac{7}{8} \cdot m_{\text{cla}}$  clauses in expectation (see, for example, Theorem 2.15 in [30]). Therefore, for any instance of MAX – E3SAT with  $m_{\text{cla}}$  clauses, the optimal objective is always at least  $\frac{7}{8} \cdot m_{\text{cla}}$ . Therefore, let  $\text{OPT}_{\text{E3SAT}}$  denote the optimal objective in MAX – E3SAT, we have

$$\frac{7}{8} \cdot m_{\text{cla}} \leq \text{OPT}_{\text{E3SAT}} \leq m_{\text{cla}}.$$

Let  $\chi$  be a constant. Based on this lower bound, we have:

$$\begin{aligned} \text{OPT}_{\text{E3SAT}} \geq \chi &\implies \text{OPT}_{\text{SVGIC}} \geq 2\chi + 6m_{\text{cla}} \\ &\geq 8\chi \\ \text{OPT}_{\text{E3SAT}} < \frac{7}{8} \cdot \chi &\implies \text{OPT}_{\text{SVGIC}} < 2 \cdot \frac{7}{8} \cdot \chi + 6m_{\text{cla}} \\ &< \frac{31}{4} \cdot \chi, \end{aligned}$$

where the second inequality holds since  $\frac{7}{8} \cdot m_{\text{cla}} \leq \text{OPT}_{\text{E3SAT}} < \frac{7}{8} \cdot \chi$  implies  $m_{\text{cla}} < \chi$ . The lemma follows.  $\square$

Therefore, combining Lemma 1 and 2, if there exists an algorithm  $\mathcal{ALG}$  that approximates SVGIC within a ratio of  $\frac{8}{31} = \frac{32}{31}$  (or equivalently, solves the  $\frac{32}{31}$ -gap version of SVGIC), then it also solves the  $\frac{8}{7}$ -gap version of MAX – E3SAT, which is known to be NP-hard. This completes the proof of the following result.

**Theorem R.1.** For any  $\epsilon > 0$ , it is NP-hard to approximate SVGIC within a ratio of  $\frac{32}{31} - \epsilon$ .

This result improves the obtained APX-hardness (i.e., the hardness of approximation is a constant  $\nabla > 1$ ). The above result is a new lower bound for  $\nabla$ . From our algorithmic

result in Theorem 5, we also know that  $\nabla \leq 4$ . Moreover, since AVG is actually a 2-approximation algorithm for the  $k = 1$  special case (please see Corollary 4.3 in Section 4.2 of the full paper [42]), it is impossible to reach  $\nabla > 2$  with  $k = 1$ . On the other hand, note that most of the closely related combinatorial maximization problems, e.g., *Maximum Happy Vertices/Edges* (MHV/MHE) [92] and *Multiway Uncut* [43] (for a detailed introduction, please see the literature review in Section 3.3 in the revised full paper [42]), have larger gaps between the proved hardness of approximation and best approximation ratios, e.g., the MHV problem [92] admits an  $O(m)$ -gap between the NP-hardness and the algorithmic result ( $O(m)$ -approximation). Furthermore, no stronger result than basic NP-hardness is known for all related MAX-problems. Our new result improves on the existing works in this regard as SVGIC is also a MAX-problem. As suggested, the above descriptions have been summarized in page 6 of the revised paper and Section 3.3 in the full paper [42].

**Reviewer 2, weak 2 (R2W2): The algorithms are simple.**

**Reply:** Thanks for the valuable comment. As suggested, we have designed an *advanced LP transformation* technique and a new *focal parameter sampling scheme* to improve the efficiency of AVG. The LP transformation technique derives a new LP formulation to reduce the number of decision variables and constraints from  $O((n+|E|)mk)$  to  $O((n+|E|)m)$  by condensing the  $x_{u,s}^c$  variables ( $k$  is the number of slots). We prove that the optimal objective in the new formulation is exactly that in the original one. The focal parameter sampling scheme maintains a *maximum utility factor*  $\bar{x}_s^{*c}$  (detailed later) for each pair of item  $c$  and slot  $s$  to avoid unnecessary sampling of focal parameters  $(c, s, \alpha)$  when  $\alpha \geq \max_{u \in V} x_{u,s}^{*c}$ , especially for a large  $k$ . We prove that the sampling results of the new sampling technique and the original one are the same. Therefore, the efficiency of AVG can be effectively improved without sacrificing the solution quality.

Moreover, we have extended SVGIC and AVG to support a series of practical scenarios. 1) *Commodity values*. Each item is associated with a commodity value to maximize the total profit. 2) *Layout slot significance*. Each slot location is associated with a different significance weight (e.g., center is better) according to retailing research [18, 77]. 3) *Multi-View Display*, where a user can be displayed multiple items in a slot, including one default personally preferred item in the primary view, and multiple items to view with friends in group views, whereas the primary and group views can be freely switched. 4) *Generalized social benefits*, where social utility can be measured not only pairwise (each pair of friends) but also group-wise (any group of friends). 5) *Subgroup change*, where the fluctuations (i.e., change of members) between the partitioned subgroups at consecutive slots are limited to ensure smooth social interactions as the elapse of time. 6) *Dynamic scenario*, where users dynamically join and leave the system with different moving speeds.

**Advanced LP Transformation.** Recall that the LP relaxation of the SVGIC problem (denote by  $\text{LP}_{\text{SVGIC}}$ ) is

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

subject to the following constraints:

$$\begin{aligned} \sum_{s=1}^k x_{u,s}^c &\leq 1, \quad \forall u \in V, c \in \mathcal{C} \\ \sum_{c \in \mathcal{C}} x_{u,s}^c &= 1, \quad \forall u \in V, s \in [k] \\ x_u^c &= \sum_{s=1}^k x_{u,s}^c, \quad \forall u \in V, c \in \mathcal{C} \\ y_e^c &= \sum_{s=1}^k y_{e,s}^c, \quad \forall e = (u, v) \in E, c \in \mathcal{C} \\ y_{e,s}^c &\leq x_{u,s}^c, \quad \forall e = (u, v) \in E, s \in [k], c \in \mathcal{C} \\ y_{e,s}^c &\leq x_{v,s}^c, \quad \forall e = (u, v) \in E, s \in [k], c \in \mathcal{C} \\ x_{u,s}^c, x_u^c, y_{e,s}^c, y_e^c &\geq 0, \quad \forall u \in V, e \in E, s \in [k], c \in \mathcal{C}. \end{aligned}$$

Solving  $\text{LP}_{\text{SVGIC}}$ , which is the first step of the AVG algorithm, deals with  $O((n + |E|)mk)$  decision variables and  $O((n + |E|)mk)$  constraints. In the following, we transform  $\text{LP}_{\text{SVGIC}}$  into a more compact linear program formulation  $\text{LP}_{\text{SIMP}}$  with fewer decision variables as follows.

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

subject to the following constraints:

$$\begin{aligned} x_u^c &\leq 1, \quad \forall u \in V, c \in \mathcal{C} \\ \sum_{c \in \mathcal{C}} x_u^c &= k, \quad \forall u \in V \\ y_e^c &\leq x_u^c, \quad \forall e = (u, v) \in E, c \in \mathcal{C} \\ y_e^c &\leq x_v^c, \quad \forall e = (u, v) \in E, c \in \mathcal{C} \\ x_u^c, y_e^c &\geq 0, \quad \forall u \in V, e \in E, c \in \mathcal{C}. \end{aligned}$$

To avoid confusion, we use normal math symbols ( $x$  and  $y$ ) to represent decision variables in  $\text{LP}_{\text{SVGIC}}$  and Typewriter symbols ( $\mathbf{x}$  and  $\mathbf{y}$ ) to denote those in  $\text{LP}_{\text{SIMP}}$ . Let  $\text{OPT}_{\text{SIMP}}$  and  $\text{OPT}_{\text{SVGIC}}$  denote the optimal objective values in  $\text{LP}_{\text{SIMP}}$  and  $\text{LP}_{\text{SVGIC}}$ . We have the following observation.

**Observation R.1.**  $\text{OPT}_{\text{SIMP}} = \text{OPT}_{\text{SVGIC}}$  always holds. Moreover, given an optimal solution  $\mathbf{X}^* = \{\mathbf{x}_u^c\}$  for  $\text{LP}_{\text{SIMP}}$ , there exists an optimal solution  $X^*$  of  $\text{LP}_{\text{SVGIC}}$  with  $x_{u,s}^c = \frac{1}{k} \cdot \mathbf{x}_u^c, \forall c, u, s$ .

*Proof.* We first prove  $\text{OPT}_{\text{SIMP}} \leq \text{OPT}_{\text{SVGIC}}$ . Given any feasible solution  $\mathbf{X}$  of  $\text{LP}_{\text{SIMP}}$ , let  $X$  denote a solution of  $\text{LP}_{\text{SVGIC}}$  such that  $x_{u,s}^c = \frac{1}{k} \cdot \mathbf{x}_u^c, \forall c, u, s$ . By construction, we have  $x_u^c = \sum_{s=1}^k x_{u,s}^c = k \cdot \frac{1}{k} \cdot \mathbf{x}_u^c = \mathbf{x}_u^c$  for all  $c, u, s$ , and analogously  $y_e^c = \mathbf{y}_e^c$  for all  $c, e$ . Therefore,  $X$  is feasible in  $\text{LP}_{\text{SVGIC}}$ . Moreover, it achieves the same objective value in  $\text{LP}_{\text{SVGIC}}$  as  $\mathbf{X}$  in  $\text{LP}_{\text{SIMP}}$ , which implies  $\text{OPT}_{\text{SIMP}} \leq \text{OPT}_{\text{SVGIC}}$ . We then prove  $\text{OPT}_{\text{SIMP}} \geq \text{OPT}_{\text{SVGIC}}$ . Given any feasible solution  $X$  of  $\text{LP}_{\text{SVGIC}}$ , let  $\mathbf{X}$  denote a solution of  $\text{LP}_{\text{SIMP}}$  such that  $\mathbf{x}_u^c = x_u^c = \sum_{s=1}^k x_{u,s}^c$ . Similarly,  $\mathbf{X}$  is a feasible solution of  $\text{LP}_{\text{SIMP}}$ , and it achieves exactly the same objective value in  $\text{LP}_{\text{SIMP}}$  as  $X$  in  $\text{LP}_{\text{SVGIC}}$ . Therefore,  $\text{OPT}_{\text{SIMP}} \geq \text{OPT}_{\text{SVGIC}}$ . Combining the above, we have  $\text{OPT}_{\text{SIMP}} = \text{OPT}_{\text{SVGIC}}$ , and the second part of the observation follows from the above construction.  $\square$

As AVG requires only *one* optimal solution for  $\text{LP}_{\text{SVGIC}}$ , it suffices for AVG to first solve the above simplified linear program  $\text{LP}_{\text{SIMP}}$  with only  $O((n + |E|)m)$  decision variables and constraints. AVG then scales the optimal solution by a factor of  $\frac{1}{k}$  according to Observation R.1 to find an optimal solution for  $\text{LP}_{\text{SVGIC}}$ . Therefore,  $\text{LP}_{\text{SIMP}}$  effectively improves the efficiency of AVG.

**Advanced Focal Parameter Sampling.** In AVG, a set of focal parameters  $(c, s, \alpha)$  is sampled uniformly at random for each iteration. However, for input instances with a large  $k$ , the utility factor  $x_{u,s}^{*c}$  of assigning item  $c$  to user  $u$  at slot  $s$  tends to become smaller, e.g., roughly  $O(\frac{1}{k})$ , especially after the above LP transformation strategy is applied. Recall that AVG compares  $\alpha$  to every user eligible for viewing item  $c$  at slot  $s$  and displays  $c$  to user  $u$  if  $x_{u,s}^{*c} \geq \alpha$ . If  $\alpha > \max_{u \in V} x_{u,s}^{*c}$ , AVG essentially remains idle (i.e., making no progress) in the current round and needs to re-sample another set of focal parameters in the next iteration. For instances with large  $k$ , it is thus more difficult for AVG to effectively sample a set of *good* focal parameters without the above property.

To address the above issue, we design an advanced focal parameter sampling scheme as follows. For each pair of  $(c, s)$ , AVG maintains the *maximum utility factor*  $\bar{x}_s^{*c} = \max\{x_{u,s}^{*c} | u \text{ is eligible for } (c, s)\}$ . Specifically,  $\bar{x}_s^{*c}$  is initialized as  $\max_{u \in V} x_{u,s}^{*c}$  after  $X^*$  (the optimal solution of LP) is retrieved, and it is constantly updated along the rounding process (as the number of users eligible for  $(c, s)$  decreases). In each iteration, AVG first randomly samples a pair  $(c, s)$  with probability  $\frac{x_s^{*c}}{\sum_{c \in \mathcal{C}, s \in [k]} x_s^{*c}}$ , i.e., proportional to the value  $\bar{x}_s^{*c}$ . It then samples  $\alpha$  uniformly at random from  $[0, \bar{x}_s^{*c}]$ .

Specifically, let **GOOD** denote the event that a set of good focal parameters is sampled, i.e.,  $\alpha$  is at most  $\bar{x}_s^{*c}$  for the selected  $(c, s)$  in the original sampling scheme. Moreover, for a set of good focal parameter  $(c, s, \alpha)$ , let **TARGET** $(c, s, \alpha)$  denote the event that Co-display Subgroup Formation ends up with the same behavior as if  $(c, s, \alpha)$  is sampled. We have the following observation. Let  $\text{Pr}_{\text{orig}}(\text{TARGET}(c, s, \alpha))$  denote the probability that **TARGET** $(c, s, \alpha)$  happens in the original sample scheme, and let  $\text{Pr}_{\text{adv}}(\text{TARGET}(c, s, \alpha))$  denote the probability of **TARGET** $(c, s, \alpha)$  in the advanced sample scheme. The following observation indicates that sampling with the advanced sampling scheme is equivalent (in terms of outcome distribution) to sampling *good* parameters, i.e., picking  $(c, s, \alpha)$  directly and only from the pool of focal parameters that lead to a successful assignment of at least one item to one user at some slot.

**Observation R.2.** For any set of good focal parameter  $(c, s, \alpha)$  such that  $\alpha \leq \max_{u \in V} x_{u,s}^{*c}$ ,  $\text{Pr}_{\text{adv}}(\text{TARGET}(c, s, \alpha)) = \text{Pr}_{\text{orig}}(\text{TARGET}(c, s, \alpha) | \text{GOOD})$ .

*Proof.* By construction,  $\text{Pr}_{\text{orig}}(\text{GOOD}) = \sum_{c \in \mathcal{C}, s \in [k]} \frac{1}{k \cdot m} (\bar{x}_s^{*c})$ .

Next, by construction, it is straightforward that the advanced sampling scheme only samples good focal parameters. Consider a set of good focal parameter  $(c, s, \alpha)$ . Let  $u_1$  and  $u_2$  denote the users such that  $u_1 = \arg \max_{x_{u,s}^{*c} < \alpha} x_{u,s}^{*c}$

and  $u_2 = \arg \min_{x_{u,s}^{*c} \geq \alpha} x_{u,s}^{*c}$ . By definition, if CSF chooses

**Algorithm R.1** AVG with advanced LP transformation and focal parameter sampling

---

**Input:**  $X^*$

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

- 1:  $\mathbf{A}(\hat{u}, \hat{s}) \leftarrow \text{NULL}$  for all  $\hat{u}, \hat{s}$
- 2: Construct  $\text{LP}_{\text{SVGIC}}$  (Original Relaxed LP)
- 3: Transform into  $\text{LP}_{\text{SIMP}}$  (Simplified LP)
- 4:  $\bar{X} \leftarrow$  optimal solution for  $\text{LP}_{\text{SIMP}}$
- 5:  $\bar{x}_s^{*c} \leftarrow \max_{u \in V} x_{u,s}^{*c}$
- 6: **for**  $u \in V, c \in \mathcal{C}, s \in [k]$  **do**
- 7:    $x_{c,s}^{*u} \leftarrow \frac{1}{k} \cdot \bar{x}_s^{*c}$
- 8: **while** some entry in  $\mathbf{A}$  is NULL **do**
- 9:   Sample  $(c, s)$  randomly w.p.  $\frac{\bar{x}_s^{*c}}{\sum_{c \in \mathcal{C}, s \in [k]} \bar{x}_s^{*c}}$
- 10:   Sample  $\alpha \in [0, \bar{x}_s^{*c}]$  uniformly at random
- 11:   **for**  $\hat{u} \in V$  **do**
- 12:     **if**  $\mathbf{A}(\hat{u}, s) = \text{NULL}$  and  $\mathbf{A}(\hat{u}, t) \neq c \forall t \neq s$  **then**
- 13:       ( $\hat{u}$  eligible for  $(c, s)$ )
- 14:       **if**  $x_{\hat{u},s}^{*c} \geq \alpha$  **then**
- 15:          $\mathbf{A}(\hat{u}, s) \leftarrow c$
- 16:     Update  $\bar{x}_s^{*c}$
- 17: **return**  $\mathbf{A}$

---

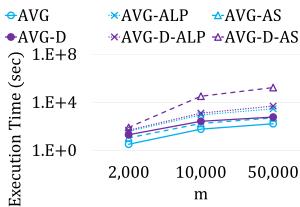


Figure R.4: Effects of speedup strategies.

any  $\alpha \in (x_{u_1,s}^{*c}, x_{u_2,s}^{*c}]$ , it ends up with the same behavior as if  $(c, s, \alpha)$  is sampled, i.e.,  $\text{TARGET}(c, s, \alpha)$  happens. With  $(c, s)$  given,  $\text{TARGET}(c, s, \alpha)$  happens with probability  $p_\alpha = x_{u_2,s}^{*c} - x_{u_1,s}^{*c}$ . Note that  $u_2$  is guaranteed to exist, and we replace  $x_{u_1,s}^{*c}$  with 0 if  $u_1$  does not exist (i.e., the case where every eligible user is assigned  $c$  by Co-display Subgroup Formation). In the advanced sampling scheme, the probability of this event is thus  $\Pr_{\text{adv}}(\text{TARGET}(c, s, \alpha)) = \frac{x_{u_2,s}^{*c}}{\sum_{c \in \mathcal{C}, s \in [k]} x_s^{*c}} \cdot p_\alpha$ . On the other hand, in the original sampling scheme, the probability of this event is  $\Pr_{\text{orig}}(\text{TARGET}(c, s, \alpha)) = \frac{1}{k \cdot m} \cdot p_\alpha$ . We have

$$\begin{aligned} \Pr_{\text{orig}}(\text{TARGET}(c, s, \alpha) | \text{GOOD}) &= \frac{\Pr_{\text{orig}}(\text{TARGET}(c, s, \alpha))}{\Pr_{\text{orig}}(\text{GOOD})} \\ &= \frac{\frac{1}{k \cdot m} \cdot p_\alpha}{\sum_{c \in \mathcal{C}, s \in [k]} \frac{1}{k \cdot m} (\bar{x}_s^{*c})} \\ &= \frac{p_\alpha}{\sum_{c \in \mathcal{C}, s \in [k]} \bar{x}_s^{*c}} \\ &= \Pr_{\text{adv}}(\text{TARGET}(c, s, \alpha)). \end{aligned}$$

The observation follows.  $\square$

Therefore, the advanced focal sampling scheme effectively improves the efficiency of AVG since it discards the non-contributing sampling events in advance. Note that it can

also be applied to improve AVG-D such that the events of selecting non-controlling focal parameters will not be simulated in the derandomization process. The main algorithm AVG with the advanced LP transformation strategy and focal sampling scheme are detailed in Algorithm R.1.

Figure R.4 shows the effects of the speedup strategies on AVG and AVG-D. AVG-ALP and AVG-D-ALP are AVG and AVG-D without the advanced LP transformation technique, respectively, while AVG-AS and AVG-D-AS are AVG and AVG-D without the advanced focal parameter sampling technique, respectively. The results manifest that all speedup strategies effectively boost the efficiency of AVG and AVG-D. For AVG, the advanced LP transformation plays a more dominant role than the advanced focal parameter sampling scheme since the computational cost of solving the linear program is more evident in AVG. Conversely, as AVG-D needs to carefully examine every combination of *good* focal parameters (i.e., those sets of parameters that do assign an item to at least one user), the advanced sampling technique has a significant effect in filtering out bad focal parameters. Therefore, the effect of the advanced LP transformation pales in comparison to that of the advanced sampling technique in AVG-D. The above has been included in Section 6.4 in the full version [42].

#### Extensions for Practical Scenarios

**A. Commodity Value.** The current objective function, i.e., maximizing the total utility among all users, is not designed from a retailer's perspective. In contrast, it will be more profitable to maximize the *total expected profit* by taking into account the commodity values, while regarding the final SAVG utility as the willingness/likelihood of purchases. Therefore, the ILP formulation of SVGIC and AVG can be modified to examine the total SAVG utility weighted by a commodity value  $\omega_c$  for each item  $c$ . Let  $\omega_c$  denote the commodity price of each item  $c$ . The new objective of SVGIC is the weighted total SAVG utility as follows.

$$\sum_{u \in V} \sum_{s=1}^k \omega_{\mathbf{A}^*(u,s)} \cdot [p(u, \mathbf{A}^*(u,s)) + \sum_{v|u \xrightarrow{\mathbf{A}^*(u,s)} v} \tau(u, v, \mathbf{A}^*(u,s))].$$

**B. Layout Slot Significance.** Retailing research [18, 77] manifests that different slot locations in the item placement layout have varying significance for users. For example, the slots at the center of an aisle are nine times more important than the slots at the ends [77]. Vertical slots right at the eye-level of users are also more significant [18]. Therefore, it is important for SVGIC to not only select the items but also optimize the corresponding placements on the shelf. Specifically, let  $\gamma_s$  represent the relative significance of slot  $s$  in the layout. The new objective of SVGIC is the SAVG utility, weighted by the relative significance as follows.

$$\sum_{u \in V} \gamma_s \cdot \sum_{s=1}^k [p(u, \mathbf{A}^*(u,s)) + \sum_{v|u \xrightarrow{\mathbf{A}^*(u,s)} v} \tau(u, v, \mathbf{A}^*(u,s))].$$

**C. Multi-View Display (MVD).** SAVG  $k$ -Configuration with CID allows each user to see only *one* item at each slot. In contrast, MVD [33, 48] has been proposed to support the display of multiple items at the same slot, so that

the primary view (for a personally preferred item) and the group view (for common items with friends) can be freely switched. Let  $\beta$  denote the maximum number of items (i.e., views) displayed for each user in each slot (depending on the user interface). an *MVD-supportive SAVG k-Configuration*  $\mathbf{A}$  maps a tuple  $(u, s)$  to an itemset  $\mathbf{A}(u, s)$  with at most  $\beta$  items, and the new objective function of SVGIC is as follows.

$$\sum_{u \in V} \sum_{s=1}^k \left( \sum_{c \in \mathbf{A}^*(u, s)} [\mathbf{p}(u, c) + \sum_{v | u \xleftrightarrow{s} v} \tau(u, v, c)] \right)$$

**D. Generalized Social Benefits.** In SVGIC  $\tau(u, v, c)$  considers only the *pair-wise* social interaction, i.e., social discussion and influence between two users viewing the common item. However, existing research on social influence [4, 36, 100] manifests that quantifying the magnitude of social influence from *group-wise* interactions is more general since the pair-wise interaction is a special case of the group-wise interaction. Therefore, let  $\tau(u, \mathcal{V}, c)$  denote the subgroup social utility for user  $u$  when co-displaying  $c$  to a *subgroup of friends*  $\mathcal{V}$  learned from group-wise social interaction models such as [36]. Let  $u \xleftrightarrow{s} \mathcal{V}$  represent the *maximal group* co-display, i.e., user  $u$  sees item  $c$  at slot  $s$  with every user in  $\mathcal{V}$ , and no friend of  $u$  outside  $\mathcal{V}$  also sees  $c$  at slot  $s$ . Thus, the new objective of SVGIC is generalized as follows.

$$\sum_{u \in V} \sum_{s=1}^k [\mathbf{p}(u, \mathbf{A}^*(u, s)) + \sum_{\mathcal{V} | u \xleftrightarrow{s} \mathcal{V}} \tau(u, \mathcal{V}, \mathbf{A}^*(u, s))].$$

Here only the maximal groups, instead of every subgroup, are examined in the objective function; otherwise, duplicated social utilities will be summed up because multiple overlapping subgroups are contained in a maximal group.

**Algorithm Extension for Scenarios A–D.** In the following, we first introduce a new ILP formulation to support the above practical scenarios. Decision variable  $x_{u,s}^c$  now indicates whether user  $u$  views item  $c$  in the *primary view* at slot  $s$ . On the other hand, let decision variable  $w_{u,s}^c$  denote whether  $u$  can see  $c$  in *some view*, i.e., in the primary view or a group view, at slot  $s$ . Therefore,  $x_{u,s}^c = 1$  implies  $w_{u,s}^c = 1$ . To consider the layout slot significance, let  $w_u^c = \sum_{s=1}^k \gamma_s \cdot w_{u,s}^c$  be the weighted sum of all  $w$  decision variables, and here  $w_u^c$  is not required to be binary. Therefore,  $w_u^c$  represents the total significance of the assigned slots for  $u$  to view  $c$ . To capture the maximal co-display, for all subgroups of users  $\mathcal{V} \subseteq V$ , let  $y_{u,\mathcal{V},s}^c$  now indicate the *maximal group* co-display. In other words,  $u \xleftrightarrow{s} \mathcal{V}$  if and only if  $y_{u,\mathcal{V},s}^c = 1$ . Analogously, let  $y_{u,\mathcal{V}}^c$  be the weighted sum of  $y_{u,\mathcal{V},s}^c$ . The new objective function is as follows.

$$\max \sum_{c \in \mathcal{C}} \omega_c \cdot \left[ \sum_{u \in V} (\mathbf{p}(u, c) \cdot w_u^c + \sum_{\mathcal{V} \subseteq V} (\tau(u, \mathcal{V}, c) \cdot y_{u,\mathcal{V}}^c)) \right]$$

subject to the constraints,

$$\sum_{c \in \mathcal{C}} x_{u,s}^c = 1, \quad \forall u \in V, s \in [k] \quad (\text{R.1})$$

$$\sum_{c \in \mathcal{C}} w_{u,s}^c \leq \beta, \quad \forall u \in V, s \in [k] \quad (\text{R.2})$$

$$x_{u,s}^c \leq w_{u,s}^c, \quad \forall u \in V, s \in [k], c \in \mathcal{C} \quad (\text{R.3})$$

$$\sum_{s=1}^k x_{u,s}^c \leq 1, \quad \forall u \in V, c \in \mathcal{C} \quad (\text{R.4})$$

$$x_u^c = \sum_{s=1}^k (\gamma_s \cdot x_{u,s}^c), \quad \forall u \in V, c \in \mathcal{C} \quad (\text{R.5})$$

$$y_{\mathcal{V}}^c = \sum_{s=1}^k (\gamma_s \cdot y_{\mathcal{V},s}^c), \quad \forall \mathcal{V} \subseteq V, c \in \mathcal{C} \quad (\text{R.6})$$

$$y_{\mathcal{V},s}^c \leq x_{u,s}^c, \quad \forall u \in \mathcal{V}, s \in [k], c \in \mathcal{C} \quad (\text{R.7})$$

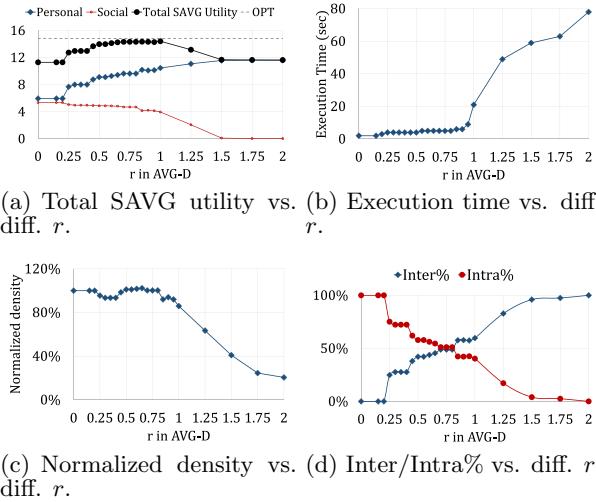
$$y_{\mathcal{V},s}^c + x_{u,s}^c \leq 1, \quad \forall u \notin \mathcal{V}, s \in [k], c \in \mathcal{C} \quad (\text{R.8})$$

$$x_{u,s}^c, y_{\mathcal{V},s}^c \in \{0, 1\}, \quad \forall u \in V, s \in [k], c \in \mathcal{C}. \quad (\text{R.9})$$

Constraint (R.1) guarantees that a user is displayed exactly one default item (for the primary view) at each slot. Constraint (R.2) ensures that at most  $\beta$  items can be viewed at one slot in MVD. Constraint (R.3) states that the default item can be viewed in the MVD, i.e., if  $c$  is displayed ( $x_{u,s}^c = 1$ ), then  $w_{u,s}^c$  will become 1. Constraint (R.4) guarantees that the default displayed items for each user are not replicated. Constraint (R.5) ensures that  $x_u^c$  (significance of the slot where  $u$  is displayed  $c$ ) is the weighted sum of  $x_{u,s}^c$ . Similarly, constraint (R.6) states that  $y_{\mathcal{V}}^c$  is the weighted sum of  $y_{\mathcal{V},s}^c$  (on the slot significance). Constraints (R.7) and (R.8) specify the maximal co-display. Constraint (R.7) states that  $y_{\mathcal{V},s}^c = 1$  only if every  $x_{u,s}^c = 1$ , implying that  $c$  is displayed to every  $u$  in  $\mathcal{V}$  at slot  $s$ . When  $y_{\mathcal{V},s}^c = 1$ , constraint (R.8) ensures no other  $x_{u,s}^c$  is 1 for  $u$  outside of  $\mathcal{V}$ , so that  $\mathcal{V}$  is indeed the maximal set. Finally, constraint (R.9) ensures the decision variables to be binary, i.e.,  $\in \{0, 1\}$ .

The LP relaxation of the above formulation also relaxes the last constraint, so that binary decision variables  $x_{u,s}^c, y_{\mathcal{V},s}^c$  become real numbers in  $[0, 1]$ . The other two phases of AVG and AVG-D remain unchanged. The commodity value  $\omega_c$  and the layout slot significance  $\gamma_s$  will not affect the theoretical guarantee of AVG and AVG-D because the original proof can be modified by replacing the preference utility and social utility values with weighted versions. Furthermore, in any iteration  $t$  in AVG, for any subgroup of users  $\mathcal{V}$  with all users  $u \in \mathcal{V}$  eligible for  $(c, s)$ , as the maximal co-displayed subgroups contain at most  $(\max |\mathcal{V}|)$  users, Lemma 7 can be analogously modified to upper bound the probability that *at least one* user in  $\mathcal{V}$  is not eligible for  $(c, s)$  in iteration  $(t+1)$  by summing exactly  $2(\max |\mathcal{V}|)$  summation terms as in the last inequality in the proof of Lemma 7. Consequently, AVG and AVG-D can achieve a  $2(\max |\mathcal{V}|)$ -approximation. Moreover, the advanced focal parameter sampling scheme can be employed in the above model.

**E. Subgroup Changes.** While the partitions of subgroups in AVG enable flexible configurations and facilitates social discussion, drastic subgroup changes between adjacent slots may undermine the smoothness of social interactions and discussions within the group when users are exploring the VE. To address this issue, one approach is to incorporate a constraint on *edit distances* between the partitioned induced subgroups at consecutive slots. Specifically, a pair of friends  $u$  and  $v$  in the same subgroup (i.e., viewing a common item) at slot  $s$  but separated into different subgroups at slot  $s+1$  will contribute 1 to the edit distance between the two slots. To reduce the subgroup changes, IP of the

**Figure R.5: Sensitivity test on  $r$  in AVG-D.**

extended AVG discourages the intermediate solution (the fractional solution) to introduce large subgroup changes by constraining the total edit distances across all consecutive slots in the SAVG  $k$ -Configuration. In the rounding scheme, AVG examines the potential outcomes before co-displaying a focal item to the target subgroup. It voids the current iteration and re-sample another set of pivot parameters, if the solution incurs a large subgroup change. Finally, after the complete SAVG  $k$ -Configuration is constructed, AVG can facilitate a local search by exchanging the sub-configuration at different slots to reduce the total subgroup change.

**F. Dynamic Scenario.** To support the dynamic join and leave of users with different walking speeds, executing the whole AVG for every dynamic event is computationally intensive. Therefore, one approach is to first solve the relaxed LP for the partial SVGIC instance with the new setting, whereas the majority of utility factors remain the same (temporarily regarded as constants in LP), so that the utility factors are updated efficiently. CSF then assigns new users to existing target subgroups according to their utility factors, and it also invokes local search to examine the potential profit from exchanging users among existing target subgroups. Moreover, similar to in SVGIC-ST, *teleportation* in VR enables a user to move directly to her friends at a different slot. Therefore, when two users are co-displayed an item (directly or indirectly) but are far apart from each other in the store, the shopping application could suggest one of them to teleport to the other's location.

**Reviewer 2, weak 3 (R2W3): Some points in experiments can be cleared further.**

**Reply:** Thanks for the comment. Please see R2D1 – R2D4.

**Reviewer 2, detailed 1 (R2D1): Is it possible and what will be the result if different members in same shopping group select different  $\lambda$ ?**

**Reply:** Yes! We agree with the reviewer that different members can be associated with different  $\lambda$  without any modification in AVG and AVG-D, because the effect of different  $\lambda$  can be achieved by re-scaling of preference (or social utility) values (i.e., already different for each member) in the objective function. For example, assume  $\lambda = 0.5$ . For a user  $u$

and her friend  $v$ , on an item  $c$ , the preference and social utility values are defined as  $w(u, c) = 0.6$  and  $\tau(u, v, c) = 0.2$ . Suppose user  $u$  instead selects  $\lambda_u = 0.8$  (favoring social utility), this effect can be captured by the following re-scaling:  $w'(u, c) = w(u, c) \cdot \frac{1-\lambda_u}{1-\lambda} = 0.24$ ,  $\tau'(u, v, c) = \tau(u, v, c) \cdot \frac{\lambda_u}{\lambda} = 0.32$ . Indeed, different users have selected their own  $\lambda$  values manually in the user study (please see Figure 16(a) in Section 16).

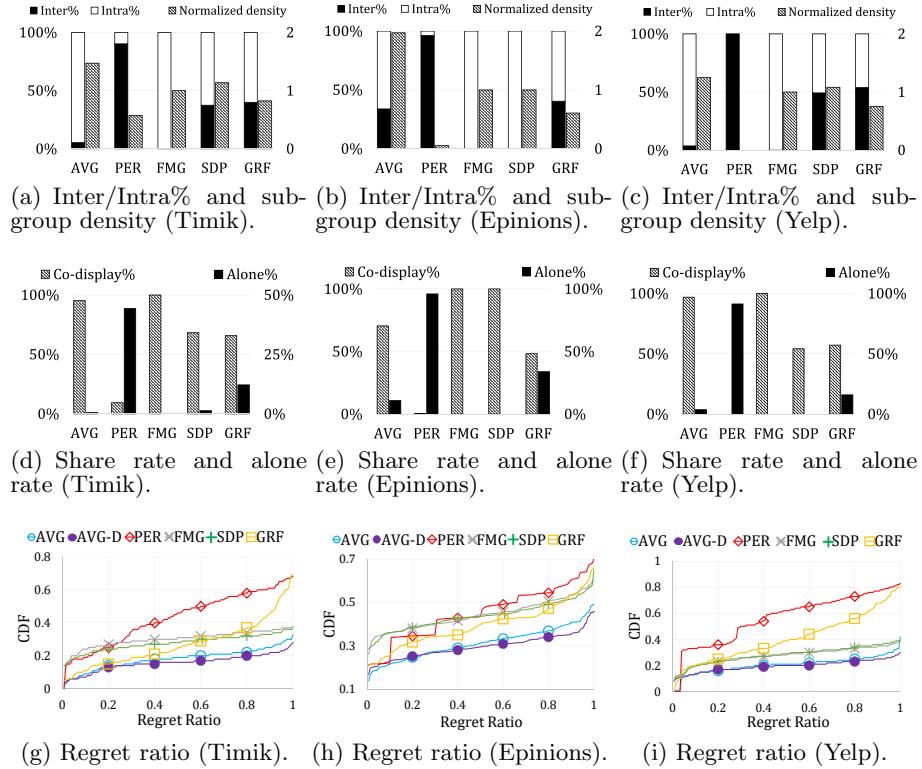
**Reviewer 2, detailed 2 (R2D2):**  $r$  seems to have a great impact on the performance. It will be good to be tested in experiment.

**Reply:** Thanks for the valuable comment. As suggested, we have incorporated additional evaluations on AVG-D with different  $r$  values as follows. We compare AVG-D with the optimal solution in the Timik dataset. Figure 5(a) shows that with  $r$  values from 0.7 to 1.0, AVG-D finds nearly-optimal solutions as Co-display Subgroup Formation with  $r$  balances between the current increment in SAVG utility and the potential SAVG utility in the future. This phenomenon is also observed in other datasets. However, setting  $r = 0.25$  is still promising as it achieves 86.1% of the optimal objective, which is consistent with the theoretical guarantee. Figure 5(a) also manifests that AVG-D with small values of  $r$  tends to resemble the group approach in Section 1 (i.e., displaying the same items to every user). For instance, AVG-D finds a large subgroup of all users with  $r \leq 0.2$ . AVG-D gradually becomes more alike to the personalized approach as  $r$  grows (e.g., the social utility is close to 0 for  $r \geq 1.5$ , implying almost every user is displayed her own favorite items). This is actually consistent with the design of AVG-D, since 1) a small  $r$  essentially ignores the potential utility gain in the future from allocating other items than the current focal item, leading to a greedy behavior of always including every eligible user into the targeted subgroup in CSF. Consequently, running every iteration with  $r$  close to 0 finds a large subgroup of all users in every slot; 2) a large  $r$  prioritizes only the future gain and thereby tends to select very few users into the targeted subgroup. This reluctant behavior in the extreme (consider  $r = \infty$ ) thus leads to subgroups consisting of only one user, essentially simulating the personalized approach. In summary, this again manifests that AVG-D can strike a good balance between the personalized and group approaches.

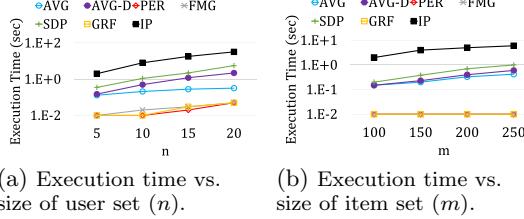
Figure 5(b) depicts the execution time of AVG-D with different values of  $r$ . Intuitively, when AVG-D chooses fewer users in the current targeted subgroup, more iterations are required to construct a complete SAVG  $k$ -Configuration. For  $n$  users and  $k$  slots, the total number of iterations is  $k$  if every targeted subgroup includes all  $n$  users but would increase to  $kn$  if every iteration displays the focal item to only one user. Therefore, AVG-D has a larger execution time with larger values of  $r$ . Figures 5(c) and 5(d) further show the average normalized density and Inter/Intra% among the subgroups in the solutions of AVG-D with different values of  $r$ . Consistent with the above discussion, the behavior of AVG-D essentially spans a spectrum from the personalized to the group approach. As suggested, the above descriptions have been included in Section 6.7 in the full paper [42].

**Reviewer 2, detailed 3 (R2D3): What are the x axis of Figures 7(b)(d)(f)? The reason why users in GRF suffer from preferences sacrificed still unclear.**

**Reply:** Thanks for the valuable comment. Note that Figures 7(b) and 7(d) are now Figures 10(g) and 10(h) due to



**Figure R.6: Comparisons on subgroup metrics (consistent with Figure 10 in the full paper [42]).**



**Figure R.7: Figures 2(c) and 2(d) in the previous version.**

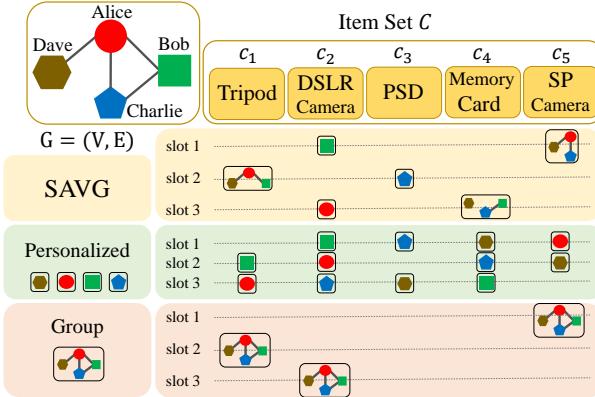
re-organization in the revised paper, and Figure 7(f) has been moved to the full version [42] as Figure 10(i). The x-axis of the figures are the *regret ratio* metric [46], which is a game-theory based measurement for the satisfaction of individual users and the overall *fairness* of the solution. A low regret ratio for a user implies that she is relatively satisfied with the SAVG  $k$ -Configuration, and thus the *fairness* among the users can be observed by inspecting the distribution of regret ratio in the final configurations. As suggested, to improve clarity, we have added the captions of the x-axis (the regret ratio) and re-organized this detailed introduction of the metric into Section 6.5, i.e., directly under Figures 10(g), 10(h), and 10(i). Moreover, we have revised the description of the metric for clarity (Please see Section 6.5 in the revised paper for detailed definitions).

As suggested, we have re-analyzed the configuration results of GRF and observed that the high regret is because of social utility sacrificed. Note that GRF forms subgroups based on preferences, i.e., users are clustered into subgroups

if they share common preferred items without considering their social relationships. The numerator of the happiness, which is the SAVG utility of a user, is lower if the user is clustered with strangers since the social utility is low even they share preferred items. Therefore, the regret is higher in this case. These results are also consistent with Figures 5 and 6. As suggested, the above discussions are added to Section 6.5 in the revised paper.

**Reviewer 2, detailed 4 (R2D4): Why the execution time of AVG, AVG-D, IP tends to have smaller slopes when  $n, m$  increase, in Figures 2 and 6?**

**Reply:** Thanks for the valuable comment. Please note that the figure numbers have changed due to re-organization of the paper. The previous Figure 2 (displayed as Figure R.7 on this page) has since been moved to Section 6.2 in the full version [42], and the previous Figure 6 is now Figure 8 in Section 6.4. AVG, AVG-D, and IP are all integer program (IP)-based approaches, which means solving an IP (or a relaxed linear program) is the first step. For a fixed  $k$  (number of display slots for a user), when  $n$  and  $m$  grow, the numbers of variables and constraints in IP and LP also increase. However, as the number of *core decisions* (which items to be displayed in the  $k$  slots) remains the same, an increasing number of newly introduced constraints become *dominated* (i.e., redundant) and thus pre-solved (relaxed) by the highly optimized commercial IP/LP solvers. Accordingly, the empirical complexity of the problem models *after* pre-solving will not grow explicitly with  $n$  and  $m$ , rendering the practical computational cost of solving the IP/LP more scalable regarding  $n$  and  $m$ .



(a) Comparison of different approaches.

SAVG	slot 1	slot 2	slot 3
Alice	c <sub>5</sub>	c <sub>1</sub>	c <sub>2</sub>
Bob	c <sub>2</sub>	c <sub>1</sub>	c <sub>4</sub>
Charlie	c <sub>5</sub>	c <sub>3</sub>	c <sub>4</sub>
Dave	c <sub>5</sub>	c <sub>1</sub>	c <sub>4</sub>

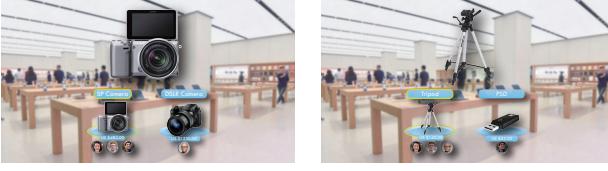
  

Person.	slot 1	slot 2	slot 3
Alice	c <sub>5</sub>	c <sub>2</sub>	c <sub>1</sub>
Bob	c <sub>2</sub>	c <sub>1</sub>	c <sub>4</sub>
Charlie	c <sub>3</sub>	c <sub>4</sub>	c <sub>2</sub>
Dave	c <sub>4</sub>	c <sub>5</sub>	c <sub>3</sub>

Group	slot 1	slot 2	slot 3
Alice	c <sub>5</sub>	c <sub>1</sub>	c <sub>2</sub>
Bob	c <sub>5</sub>	c <sub>1</sub>	c <sub>2</sub>
Charlie	c <sub>3</sub>	c <sub>4</sub>	c <sub>2</sub>
Dave	c <sub>4</sub>	c <sub>5</sub>	c <sub>3</sub>

(b) Item assignments of different approaches.



(c) Alice's view at slot 1.

(d) Alice's view at slot 2.

**Figure R.8: Illustrative example.**

Moreover, in AVG and AVG-D, after the optimal fractional solution for the LP is retrieved, decision variables with values of 0 or 1 do not require the rounding procedure and can be directly filled into the final solution. This usually happens when 1) some user has a very strong preference toward a specific item such that viewing the item on her own still outweighs viewing other items with friends, and 2) some item is extremely popular and thus is co-displayed to a majority of users. As  $n$  and  $m$  increases in this case, the total number of decision variables  $x^*(u, \cdot)$  grows, but the summation of all decision variables remains constant in Constraint (2). Therefore, more decision variables have values close to 0 in this case. On the other hand, all other baselines are based on local greedy heuristics that need either a linear scan on all users ( $O(n)$ ) or all items ( $O(m)$ ) at each step, and do not benefit from the above “decision dilution” phenomenon. As suggested, the above discussions are included in Section 6.4 in the full paper [42].

**Reviewer 4, deetailed weak 1.1 (R4DW1.1):** The differences between direct display and indirect display are not clearly explained. Based on the explanation in this paper, two users with direct display have the same item showing on the same slot and users with indirect display have the same item displaying on different slots. It seems like the differences between direct display and indirect display are not much because users can easily see all three items on different slots (based on Figures 1 (b) and (c)).

**Reply:** Thanks for the valuable comment. For the difference between these notions, please see the related response for R1D3.

**Reviewer 4, deetailed weak 1.2 (R4DW1.2):** In section 4 of the full-version paper, it says that “By relaxing all the integrality constraints in the IP” on page 8. However, how to relax the constraints is not clearly explained. That is an important step and needs to be better explained.

**Reply:** Thanks. We follow the standard linear relaxation technique [86] to relax an ILP problem into an LP problem. Specifically, Constraint (7) in the ILP associates an integrality constraint with every decision variable, i.e.,  $x_{u,s}^c, x_u^c, y_{e,s}^c, y_e^c \in \{0, 1\}$ , to restrict every decision variable to a binary value. The LP relaxation is formulated by replacing the above constraint with a linear constraint, i.e.,  $0 \leq x_{u,s}^c, x_u^c, y_{e,s}^c, y_e^c \leq 1$ , such that every decision variable is a real value between 0 and 1, and the relaxed problem can be solved by commercial solvers, e.g., CPLEX or Gurobi. As suggested, the above explanation has been included in Section 4.1 in the revised paper.

**Reviewer 4, deetailed weak 1.3 (R4DW1.3):** Three decision variables,  $x$ ,  $y$ , and  $z$ , are defined (see formula 10 in the full-version paper) in Section 3, but only  $x$  is used in the proposed AVG algorithm. What about  $y$  and  $z$ ?

**Reply:** Thanks. Variables  $x$  are sufficient to represent the solution of SVGIC (i.e.,  $x_{u,s}^c$  denotes whether item  $c$  is displayed at slot  $s$  for user  $u$ ), whereas variables  $y$  and  $z$  are only auxiliary variables that enable SVGIC to be formulated as ILP. Without incorporating  $y$  and  $z$ , the formulation will become *non-linear*. The reason is that  $\min\{x_{u,s}^c, x_{v,s}^c\}$  can represent whether users  $u$  and  $v$  are co-displayed item  $c$  at slot  $s$ , i.e., 1 for the co-display case. Similarly,  $\min\{x_u^c, x_v^c\} = 1$  represents that users  $u$  and  $v$  both see item  $c$  at some (not necessarily the same) slots. However, the formulations will become non-linear if the above terms appear in the objective function and constraints. Therefore, the design of  $y$  and  $z$  variables is only a formulation technique to make SVGIC linear. On the other hand, any combinatorial algorithm does not need to follow the above linear requirement (i.e., AVG can directly find  $\min\{x_{u,s}^c, x_{v,s}^c\}$ ). Variables  $y$  and  $z$  thereby are not considered in AVG because  $x$  is sufficient to represent the solution. As suggested, the above explanation has been included in Section 3.3 and 4.1 in the revised paper.

**Reviewer 4, detailed weak 2.1 (R4DW2.1):** Some tables/figures are far away from their descriptions, which makes this paper harder to follow. In section 1 (page 2), Table 1 is used to explain Figure 1. However, Table 1 is placed in Section 3 (page 5). Figure 1 and Table 3 are far away from each other. Readers have to go backward and forward to understand the description. Figure 1 (a) shows each item displaying to different users, one column in Figure 1 (a) shows one item. However, if I want to compare the different items showing to different users (for example, Alice and Bob), Figure 1 (a) is not intuitive enough to show the differences. It will be better if there is one subfigure showing the differences between different users.

**Reply:** Thanks for the valuable comments! As suggested,

we have re-organized the paper to make sure the tables and figures are close to their descriptions. For the motivation example shown as Figure R.8 here, we have added more subfigures that directly show the item assignment for each user by each approach. Moreover, we have modified the corresponding descriptions to make sure that the comprehension of the example does not rely on Table 1. The revised illustrative example is shown as Figure R.8.

**Illustrative Example (Consistent with Example 1).** Figure 8(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. Figure 8(b) shows the same configuration as Figure 8(a) by depicting the individual item assignments for each user with different approaches.

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 (which is consistent to the numerical example 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 8(c) and 8(d) 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. □

**Reviewer 4, detailed weak 2.2 (R4DW2.2):** This paper is a short version of the full-version paper. However, some important information is missing in the short-version paper, which makes the paper hard to follow. The introduction of this paper mentions that “We define the notion of indirect co-display to capture...” (page 3). However, “indirect co-display” is not defined in this short-version paper, although it is mentioned in the full-version paper. The concept of “indirect co-display” is important and it should be explained in the short-version paper too. Without showing the meaning of some symbols, it is hard for the reader to understand the formulas. For example,  $N_P(S)$  is used in Definition 2. However, the meaning of  $N_P(s)$  is only explained in Table 2 (page 5) in the full-version paper.

**Reply:** Thanks for the valuable comments. As suggested, we have re-organized the paper and moved the definition of the indirect co-display to the short-version paper, and checked the interdependency between the short-version paper and the full-version paper. Also, we have added the meaning of  $N_P(s)$  into Definition 2.

# 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 APX-hard and also NP-hard to approximate within  $\frac{32}{31} - \epsilon$ . 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% 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*, 14(xxx):xxxx-yyyy, 2020.

DOI:

## 1. INTRODUCTION

Virtual Reality (VR) has emerged as a disruptive technology for social [23], travel [82], and E-commerce applications. Particularly, a marketing report about future retail from Oracle [8] manifests that 78% of online retailers already have implemented or are planning to implement VR and AI. Recently, International Data Corporation (IDC) forecasts the worldwide spending on VR/AR to reach 18.8 billion USD in 2020 [87], including \$1.5 billion in retail.

This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>. For any use beyond those covered by this license, obtain permission by emailing info@vldb.org. Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.

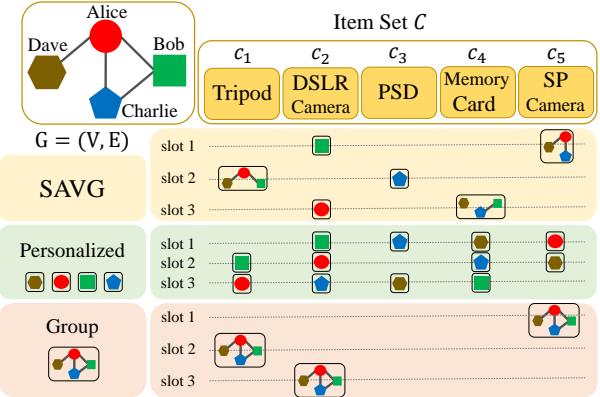
*Proceedings of the VLDB Endowment*, Vol. 14, No. xxx  
ISSN 2150-8097.

DOI:

It also foresees the VR/AR market to continue an annual growth rate of 77% through at least 2023. Moreover, 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 [80], eBay and Myer's VR department store [78], Alibaba Buy+ [2], and IKEA VR Store [81]. Although these VR shopping platforms look promising, most of them are designed only for a single user instead of a group of friends, who 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 [91, 94, 95]. 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.

The CID technology [33, 48] naturally enables VR group shopping systems with two unique strengths: 1) *Customization*. IKEA (see video [81] at 0:40) and Lowe's [49] respectively launch VR store applications where the displayed furniture may adapt to the preferences of their users, which attracts more than 73% of the users to recommend IKEA VR store [38]. Alibaba's [24] and eBay's VR stores [53] also devote themselves to provide personalized product offers. According to a marketing survey, 79% of surveyed US, UK, and China consumers are likely to visit a VR store displaying customized products [25]. 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 [94, 95]. As a result, the displayed items could be tailored to maximize potential discussions during group shopping. A survey of 1,400 consumers in 2017 shows that 65% of the consumers are excited about VR shopping, while 54% of them acknowledge social shopping as their ways to pur-



(a) Comparison of different approaches.

	slot 1	slot 2	slot 3
<b>SAVG</b>			
Alice	c <sub>5</sub>	c <sub>1</sub>	c <sub>2</sub>
Bob	c <sub>2</sub>	c <sub>1</sub>	c <sub>4</sub>
Charlie	c <sub>5</sub>	c <sub>3</sub>	c <sub>4</sub>
Dave	c <sub>5</sub>	c <sub>1</sub>	c <sub>4</sub>

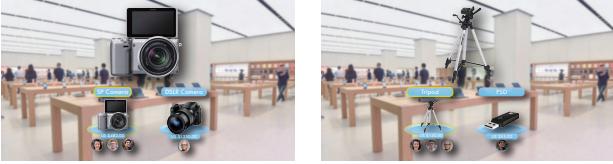
  

	slot 1	slot 2	slot 3
<b>Person.</b>			
Alice	c <sub>5</sub>	c <sub>2</sub>	c <sub>1</sub>
Bob	c <sub>2</sub>	c <sub>1</sub>	c <sub>4</sub>
Charlie	c <sub>3</sub>	c <sub>4</sub>	c <sub>2</sub>
Dave	c <sub>4</sub>	c <sub>5</sub>	c <sub>3</sub>

	slot 1	slot 2	slot 3
<b>Group</b>			
Alice	c <sub>5</sub>	c <sub>1</sub>	c <sub>2</sub>
Bob	c <sub>5</sub>	c <sub>1</sub>	c <sub>2</sub>
Charlie	c <sub>5</sub>	c <sub>1</sub>	c <sub>2</sub>
Dave	c <sub>5</sub>	c <sub>1</sub>	c <sub>2</sub>

(b) Item assignments of different approaches.



(c) Alice's view at slot 1.

(d) Alice's view at slot 2.

Figure 1: Illustrative example.

chase products [58]. The L.E.K. consulting survey [34] shows that 70% of 1,000 consumers who had already experienced VR technology are strongly interested in virtual shopping with friends who are not physically present. Embracing the trend, Shopify [74] and its technical partner Qbit [59] build a social VR store supporting attendance of multiple users with customized display [1]. 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.

Encouraged by the above evidence, we make the first attempt to formulate the 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 (and its subgroups). For customization, one possible approach is to use existing personalized recommendation techniques [12,31,32] 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 sys-

tems [35, 60, 67, 71] 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. Figure 1(b) shows the same configuration as Figure 1(a) by depicting the individual item assignments for each user with different approaches.

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 (which is consistent with the numerical example 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(c) and 1(d) 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.  $\square$

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 *Social-aware VR Group-Item Configuration* (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. The problem is very challenging due to a complex trade-off between prioritizing personalized item configuration and facilitating social interactions. Indeed, we prove that SVGIC is NP-hard to approximate SVGIC within a ratio of  $\frac{32}{31} - \epsilon$ .

To solve SVGIC, we first present an Integer Program (IP) as a baseline to find the exact solution which requires super-polynomial time. To address the efficiency issue while ensuring good solution quality, we then propose a novel approximation algorithm, namely *Alignment-aware VR 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. We design further LP transformation and sampling techniques to improve the efficiency of AVG.

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* [6] 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. Moreover, we have extended AVG to support a series of practical scenarios. 1) *Commodity values*. Each item is associated with a commodity value to maximize the total profit. 2) *Layout slot significance*. Each slot location is associated with a different significance (e.g., center is better) according to

retailing researches [18, 77]. 3) *Multi-View Display*, where a user can be displayed multiple items in a slot, including one default personally preferred item in the *primary view*, and multiple items to view with friends in *group views*, whereas the primary and group views can be freely switched. 4) *Generalized social benefits*, where social utility can be measured from not only the pairwise (each pair of friends) but also group-wise (any group of friends) perspectives. 5) *Subgroup change*, where the fluctuations (i.e., change of members) between the partitioned subgroups at consecutive slots are limited to ensure smooth social interactions as the elapse of time. 6) *Dynamic scenario*, where users dynamic join and leave the system with different moving speeds. Finally, we have also identified *Social Event Organization* (SEO) as another important application of the targeted problem.

The contributions of this work are summarized 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 and also NP-hard to approximate within  $\frac{32}{31} - \epsilon$ .
- 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 discusses the directions for extensions. Section 6 reports the experimental results, and Section 7 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 [60, 67]. Cao *et al.* [9] 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 [65] forms subgroups according to item preferences. SDSSel [71]

finds dense subgroups and diverse itemsets. Shuai *et al.* [75] 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.

**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) [12, 51] on heterogeneous applications [26], while Bayesian Personalized Ranking [11, 63] is proposed to learn the ranking of recommended items. However, the above works fail to consider social interactions. While social relations have been leveraged to infer preferences of products [98, 99], POIs [44], and social events [47], they do not take into account the social interactions among users in recommendation of items. Thus, they fail to consider the trade-off between social and personal factors. 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 [13, 45]. Group formation organizes a group of experts with low communication costs and specific skill combinations [5, 61]. In addition, organizing a group in location-based social networks based on certain spatial factors has also gained more attention [72, 73]. 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.

**Related Combinatorial Optimization Problems.** SVGIC is related to the *Multi-Labeling* (ML) [88] problem and its variations, including Multiway Partition [88, 96], *Maximum Happy Vertices/Edges* (MHV/MHE) [92], and *Multiway Cut* [16] in graphs. In Section 3.4, we revisit the challenging combinatorial nature of the proposed SVGIC problem and its relation with the related problems with detailed introduction of all problems and a summary table.

### 3. PROBLEM FORMULATION AND HARDNESS RESULTS

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  $\mathbf{A}(\cdot, \cdot) : (V \times [k]) \rightarrow \mathcal{C}$  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  $\mathbf{A}(u, :) = \langle \mathbf{A}(u, 1), \mathbf{A}(u, 2), \dots, \mathbf{A}(u, k) \rangle$  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 [91, 94, 95] demonstrates that discussions and interactions are inclined to trigger more purchases of an item. Specifically, the VR technology supporting social interactions has already been employed in existing VR social platforms, e.g., VRChat [83], AltspaceVR [3], and Facebook Horizon [22], where users, represented by customized avatars, interact with friends in various ways: 1) *Movement and gesture*. Commercial VR devices, e.g., Oculus Quest and HTC Vive Pro, support six-degree-of-freedom headsets and hand-held controllers to detect the user movements and gestures in real-time. These movements are immediately updated to the server and then perceived by other users. For example, a high-five gesture mechanism is used in Rec Room [62] to let users befriend others. 2) *Emotes and Emojis*. Similar to emoticons in social network platforms, VR users can present their emotions through basic expressions (happy face, sad face, etc.) or advanced motions (e.g., dance, backflip, and die in VRChat [84]). 3) *Real sound*. This allows users to communicate with friends verbally through their microphones. As such, current VR applications support immersive and seamless social interactions, rendering social interactions realistic in shopping systems.

With non-customized user environments in brick-and-mortar stores, discussing a specific item near its location is the default behavior. On the other hand, as users are displayed completely customized items in personalized e-commerce (e.g., the 20 items appearing on the main page of a shopping site can be completely disjoint for two friends), it often requires a “calibration” step (e.g., sharing the exact hyperlink of the product page) to share the view on the target item before e-commerce users can engage in discussions on some items. Since shopping users are used to having discussions on the common item they are viewing at the identical slot in their own Virtual Environments (VEs), we devote to enhance the possibility of this intuitive setting in our envisioned VR shopping VEs even though other settings are also possible in VR.

Therefore, for a pair of friends displayed a common item, it is most convenient to display the item at the same slot in the two users’ respective VEs such that they can locate the exact position easily (e.g., “come to the second shelf; you should see this.”). Accordingly, upon viewing any item, it is expected that the user interface indicates a list of friends being co-displayed the specific item with the current user, so that she is aware of the potential candidates to start a

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

	$p(u_A, \cdot)$	$p(u_B, \cdot)$	$p(u_C, \cdot)$	$p(u_D, \cdot)$	$\tau(u_A, u_B, \cdot)$	$\tau(u_A, u_C, \cdot)$	$\tau(u_A, u_D, \cdot)$	$\tau(u_B, u_A, \cdot)$	$\tau(u_B, u_C, \cdot)$	$\tau(u_B, u_D, \cdot)$	$\tau(u_C, u_A, \cdot)$	$\tau(u_C, u_B, \cdot)$	$\tau(u_C, u_D, \cdot)$	$\tau(u_D, u_A, \cdot)$
$c_1$	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	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	0.1	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	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	0.05	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
$p(u, c)$	preference utility
$\tau(u, v, c)$	social utility
$w_{\mathbf{A}}(u, c)$	SAVG utility
$\lambda$	weight of social utility
$p'(u, c)$	scaled preference utility
$u \xrightarrow[s]{} v$	Co-display (at slot $s$ )
$u \xrightarrow[c]{} v$	Co-display (at some slot)
$N_p(s)$	number of user subgroups at slot $s$
$V^s$	collection of subgroups at slot $s$
$x_{u,s}^c$	decision variable of $u$ displayed $c$ at slot $s$
$x_u^c$	decision variable of $u$ displayed $c$
$y_{e,s}^c$	decision variable of $u, v$ co-displayed $c$ at slot $s$
$y_e^c$	decision variable of $u, v$ co-displayed $c$

discussion with. 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* ( $u \xrightarrow[s]{} v$ ). Let  $u \xrightarrow[s]{} v$  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  $u \xrightarrow[c]{} v$  denote that there exists at least one  $s$  such that  $u \xrightarrow[s]{} v$ .

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 disjoint subsets  $V^s = \{V_1^s, V_2^s, \dots, V_{N_p(s)}^s\}$ , where  $N_p(s)$  is the number of partitioned subgroups at slot  $s$ , such that  $u \xrightarrow[s]{} v$  if and only if  $u, v \in V_i^s, i = 1, \dots, N_p(s)$ .

Under the scenario of brick-and-mortar group shopping, previous research [50, 52, 66, 91, 94] indicates that the satisfaction of a group shopping user is affected by two factors: *personal preference* and *social interaction*. Further-

more, empirical research on dynamic group interaction in retail finds that shopping while discussing with others consistently boosts sales [91], while intra-group talk frequency has a significant impact on users' shopping tendencies and purchase frequencies [94, 95]. Accordingly, we capture the overall satisfaction by combining 1) the aggregated personal preferences and 2) the retailing benefits from facilitating social interactions and discussions. We note that, from an optimization perspective, these two goals form a trade-off because close friends/families may have diverse personal preferences over items. One possible way to simultaneously consider both factors is to use an end-to-end machine learning-based approach, which generates the *user embeddings* and *item embeddings* [9] and designs a neural network aggregator to derive the total user satisfaction. However, this approach requires an algorithm to generate candidate configurations for ranking and relies on a huge amount of data to train the parameters in the aggregator. On the other hand, previous research [76, 85] has demonstrated that a *weighted combination* of the preferential and social factors is effective in measuring user satisfaction, where the weights can be directly set by a user or implicitly learned from existing models [47, 97]. Another possible objective is a weighted sum of the previous objective metric (the aggregated preference) with the aggregated social satisfaction as the constraint, or vice versa, which is equivalent to our problem after applying Lagrangian relaxation to the constraint. Therefore, we follow [76, 85] to define the SAVG utility as a combination of *aggregated preference utility* and *aggregated social utility*, weighted by a parameter  $\lambda$ .

Specifically, for a given pair of friends  $u$  and  $v$ , i.e.,  $(u, v) \in E$ , and an item  $c \in \mathcal{C}$ , let  $p(u, c) \geq 0$  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  $v$ , where  $\tau(u, v, c)$  can be different from  $\tau(v, u, c)$ . Following [76, 85], the *SAVG utility* is defined as follows.

**Definition 3.** *SAVG utility* ( $w_{\mathbf{A}}(u, c)$ ). 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.

$$w_{\mathbf{A}}(u, c) = (1 - \lambda) \cdot p(u, c) + \lambda \cdot \sum_{v|u \xrightarrow[c]{} v} \tau(u, v, c)$$

The preference and social utility values can be directly given by the users or obtained from social-aware personalized recommendation learning models [47, 97]. 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. The weight  $\lambda$  can also be directly set by a user or implicitly learned from existing models [47, 97]. To validate the effectiveness of this objective model, we have conducted a

**Table 3: Notations used in SVGIC-ST.**

Symbol	Description
$w_A(u, c)$	SAVG utility with indirect co-display
$u \xrightarrow[s, s']^c v$	Indirect Co-display (at slots $s, s'$ )
$u \xrightarrow[\text{ind}]^c v$	Indirect Co-display (at some slots)
$d_{\text{tel}}$	discount factor
$M$	subgroup size constraint
$z_e^c$	decision variable of $u, v$ co-displayed $c$

user study in real CID VR shopping system which shows high correlations between user satisfaction and the proposed problem formulation (Please see Section 6.9 for the results).

**Example 2.** Revisit Example 1 where  $\mathcal{C} = \{c_1, c_2, \dots, c_5\}$  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  $p(u_A, c_1) = 0.8$ . 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,  $w_A(u_A, c_1) = 0.6 \cdot 0.8 + 0.4 \cdot (0.2 + 0.2) = 0.64$ .  $\square$

We then formally introduce the SVGIC problem as follows. All used notations in SVGIC are summarized in Table 2.

**Problem:** Social-aware VR Group-Item Configuration.

**Given:** A social network  $G = (V, E)$ , a universal item set  $\mathcal{C}$ , preference utility  $p(u, c)$  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  $\lambda$ , and the number of slots  $k$ .

**Find:** An SAVG  $k$ -Configuration  $\mathbf{A}^*$  to maximize the total SAVG utility

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

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

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  $\lambda$  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,  $\mathbf{A}(u, s) = \mathbf{A}(v, s), \forall u, v \in V, s = 1, \dots, k$ , 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  $\text{OPT}(\mathbf{I})$  denote

the optimal objective value of SVGIC. Let  $\text{OPT}_P(\mathbf{I})$  and  $\text{OPT}_G(\mathbf{I})$  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 1.** Given any  $n$ , there exists 1) a SVGIC instance  $\mathbf{I}_G$  with  $n$  users such that  $\frac{\text{OPT}(\mathbf{I}_G)}{\text{OPT}_G(\mathbf{I}_G)} = n$ . 2) a SVGIC instance  $\mathbf{I}_P$  such that  $\frac{\text{OPT}(\mathbf{I}_P)}{\text{OPT}_P(\mathbf{I}_P)} \geq \frac{\lambda}{1-\lambda} \cdot \frac{n-1}{2} = O(n)$ .

*Proof.* We construct  $\mathbf{I}_G$  and  $\mathbf{I}_P$  as follows. For  $\mathbf{I}_G$ , each user  $u_i$  in  $\mathbf{I}_G$  prefers exactly  $k$  items  $\mathcal{C}_i = \{c_i, c_{n+i}, \dots, c_{(k-1)n+i}\}$ , i.e.,  $p(u_i, c_j) = 1$  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  $\frac{\text{OPT}(\mathbf{I}_G)}{\text{OPT}_G(\mathbf{I}_G)} = \frac{k(1-\lambda)}{k(1-\lambda)} = n$ .

For  $\mathbf{I}_P$ , each user  $u_i$  in  $\mathbf{I}_P$  prefers the items in  $\mathcal{C}_i$  only *slightly more than* all the other items. Specifically, let  $p(u_i, c_j) = 1$  for  $c_j \in \mathcal{C}_i$ , and  $p(u_i, c_j) = 1 - \epsilon$  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  $\epsilon$  approaches 0,  $\frac{\text{OPT}(\mathbf{I}_P)}{\text{OPT}_P(\mathbf{I}_P)} \geq \frac{k[(1-\lambda)\cdot((1-\epsilon)n+\epsilon)+\lambda \cdot \frac{n(n-1)}{2}]}{(1-\lambda) \cdot n \cdot k} \approx 1 + \frac{\lambda}{1-\lambda} \cdot \frac{n-1}{2} = O(n)$ , regarding  $\lambda$  as a constant.  $\square$

## 3.2 Indirect Co-display and SVGIC-ST

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*. On the other hand, if a common item is displayed at different slots in two users' VEs, a prompt discussion is more difficult to initiate since the users need to locate and move to the exact position of the item first. *Teleportation* [6], 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 discussion. To model this event that requires more efforts from users, we further propose a generalized notion of *indirect co-display* to consider the above-described scenario.

**Definition 4.** *Indirect Co-display.*  $(u \xrightarrow[s, s']^c v)$ . Let  $u \xrightarrow[\text{ind}]^c v$  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  $u \xrightarrow[\text{ind}]^c v$  to denote that there exist different slots  $s \neq s'$  such that  $u \xrightarrow[s, s']^c v$ .

Note that we use  $u \leftrightarrow^c v$  to denote *direct co-display*, i.e., there exists at least one  $s$  such that  $u \xleftrightarrow[s]^c v$ , while using

$u \xrightarrow[\text{ind}]{c} v$  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:  $u \xrightarrow[s]{c} v$ .

As social discussions on indirectly co-displayed items are less immersive and require intentional user effort, we introduce a discount factor  $d_{\text{tel}} < 1$  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* ( $w'_{\mathbf{A}}(u, c)$ ). 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

$$w'_{\mathbf{A}}(u, c) = (1 - \lambda) \cdot p(u, c) + \lambda \cdot \left( \sum_{v|u \xleftrightarrow[c]{} v} \tau(u, v, c) + \sum_{v|u \xleftrightarrow[\text{ind}]{c} v} d_{\text{tel}} \cdot \tau(u, v, c) \right).$$

Finally, we note that existing VR applications often do not accommodate an unlimited number of users in one shared VE, e.g., VRChat [83] allows at most 16 users, and IrisVR [39] 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  $C$ , preference utility  $p(u, c)$  for all  $u$  and  $c$ , social utility  $\tau(u, v, c)$  for all  $u, v$ , and  $c$ , the weight  $\lambda$ , the number of slots  $k$ , the teleportation discount  $d_{\text{tel}}$ , and the subgroup size upper bound  $M$ .

**Find:** An SAVG  $k$ -Configuration  $\mathbf{A}^*$  to maximize the total SAVG utility (with indirect co-display)

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

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

Notations in SVGIC-ST are summarized in Table 3.

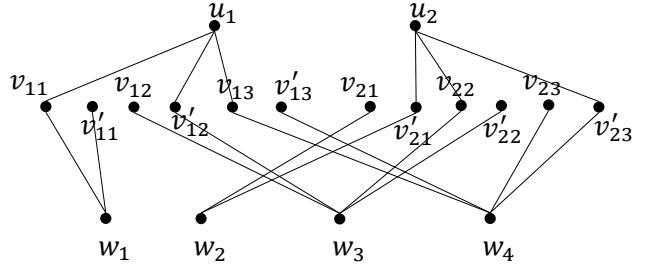
### 3.3 Hardness Result and Integer Program

In the following, we prove that SVGIC is APX-hard and also NP-hard to approximate within a ratio of  $\frac{32}{31} - \epsilon$ . We then prove that SVGIC-ST does not admit any constant-factor polynomial-time approximation algorithms, unless the Exponential Time Hypothesis (ETH) fails.

**Theorem 2.** SVGIC is APX-hard. Moreover, for any  $\epsilon > 0$ , it is NP-hard to approximate SVGIC within a ratio of  $\frac{32}{31} - \epsilon$ .

In the following, we prove the hardness result with a gap-preserving reduction from the MAX – E3SAT problem [30].

$$\phi = (a_1 \vee \bar{a}_3 \vee a_4) \wedge (\bar{a}_2 \vee a_3 \vee \bar{a}_4)$$



**Figure 2:** An illustration instance built for MAX – E3SAT

Given a conjunctive normal form (CNF) formula  $\phi$  with exactly 3 literals in each clause, MAX – E3SAT asks for a truth assignment of all variables such that the number of satisfied clauses in  $\phi$  is maximized. The following lemma (from [30]) states the hardness of approximation of MAX – E3SAT.

**Lemma 1.** MAX – E3SAT is NP-hard to approximate within a ratio of  $\frac{8}{7} - \epsilon$ .

Our reduction builds on the following lemma.

**Lemma 2.** There exists a gap-preserving reduction from MAX – E3SAT to SVGIC which translates from a formula  $\phi$  to an SVGIC instance  $\mathbf{I}$  such that

- If at least  $\chi$  clauses can be satisfied in  $\phi$ , then the optimal value of  $\mathbf{I}$  is at least  $8 \cdot \chi$ ,
- If less than  $\frac{7}{8} \cdot \chi$  clauses can be satisfied in  $\phi$ , then the optimal value of  $\mathbf{I}$  is less than  $\frac{31}{4} \cdot \chi$ .

*Proof.* Given a MAX – E3SAT instance  $\phi$  with  $n_{\text{var}}$  Boolean variables  $a_1, \dots, a_{n_{\text{var}}}$  and  $m_{\text{cla}}$  clauses  $C_1, \dots, C_{m_{\text{cla}}}$ , we construct an SVGIC instance as follows. 1) Each clause  $C_j = (l_{j1} \vee l_{j2} \vee l_{j3})$  corresponds to i) a vertex  $u_j \in V_1$ , and ii) six vertices  $v_{j1}, v'_{j1}, v_{j2}, v'_{j2}, v_{j3}, v'_{j3} \in V_2$ , where  $v_{jt}$  corresponds to  $a_i$ , and  $v'_{jt}$  corresponds to  $\bar{a}_i$  (the negation of  $a_i$ ), if and only if  $l_{jt}$  is  $a_i$  or  $\bar{a}_i$ . 2) Each Boolean variable  $a_i$  corresponds to a vertex  $w_i \in V_3$ . (Thus,  $V = (V_1 \cup V_2 \cup V_3)$  has  $(n_{\text{var}} + 7 \cdot m_{\text{cla}})$  vertices.) 3) For each clause  $C_j$ , we add an edge between  $u_j$  and each of the three vertices in  $V_2$  that correspond to TRUE assignments of the literals in  $C_j$ . 4) For each literal  $l_{jt}$ , if  $l_{jt}$  is  $a_i$  or  $\bar{a}_i$ , we add an edge between  $w_i$  and  $v_{jt}$  and also an edge between  $w_i$  and  $v'_{jt}$ . (Therefore,  $G$  has  $(9 \cdot m_{\text{cla}})$  edges.) 5) For each edge between  $V_1$  and  $V_2$  with the form  $(u_j, v_{jt})$ , we add an item  $c_{jt}$ ; similarly, for each edge between  $V_1$  and  $V_2$  with the form  $(u_j, v'_{jt})$ , we add an item  $c'_{jt}$ . 6) For each Boolean variable  $a_i$ , we add two items  $c_i$  and  $c'_i$ . 7) For the preference utility values, let  $p(v, c) = 0$  for all  $v \in V, c \in C$ . 8) For the social utility values, let  $\tau(u_j, v_{jt}, c_{jt}) = \tau(v_{jt}, u_j, c_{jt}) = 1$  for every  $(u_j, v_{jt}) \in E$ ; similarly, let  $\tau(u_j, v'_{jt}, c'_{jt}) = \tau(v'_{jt}, u_j, c'_{jt}) = 1$  for every  $(u_j, v'_{jt}) \in E$ . Moreover, let  $\tau(w_i, v_{jt}, c_i) = \tau(v_{jt}, w_i, c_i) = 1$  if and only if  $l_{jt} = a_i$ ; similarly, let  $\tau(w_i, v'_{jt}, c'_i) = \tau(v'_{jt}, w_i, c'_i) = 1$  if and only if  $l_{jt} = \bar{a}_i$ . All other  $\tau$  values are zero. An example is illustrated in Figure 2. Finally, let  $k = \lambda = 1$ .

We first prove the sufficient condition. Assume that there exists a truth assignment in  $\phi$  satisfying at least  $m$  clauses. We construct a feasible solution for SVGIC as follows. 1) For each satisfied clause  $C_j$ , let  $\underline{t}_j$  be the smallest  $t$  such that  $1_{jt}$  is assigned **TRUE**.  $u_j$  is displayed item  $c_{jt}$  if  $1_{jt}$  is of the form  $a_i$ ; otherwise  $u_j$  is displayed item  $c'_{jt}$ ; 2) For each literal  $1_{jt}$  with the form  $a_i$  and assigned **TRUE**,  $v_{jt}$  is displayed item  $c_{jt}$ ; for each literal  $1_{jt}$  with the form  $\bar{a}_i$  and assigned **TRUE**,  $v'_{jt}$  is displayed item  $c'_{jt}$ ; 3) For each literal  $1_{jt}$  with the form  $a_i$  assigned **FALSE**,  $v_{jt}$  is displayed item  $c_i$ ; for each literal  $1_{jt}$  with the form  $\bar{a}_i$  assigned **FALSE**,  $v'_{jt}$  is displayed item  $c'_i$ ; 4) For each Boolean variable  $a_i$ ,  $w_i$  is displayed item  $c_i$  if and only if  $a_i$  is **FALSE** in the assignment; otherwise  $w_i$  is displayed item  $c'_i$ ; 5) For an unsatisfied clause  $C_j$ ,  $u_j$  can be displayed any item.

In the following, we prove that this solution achieves an objective of at least  $(2 \cdot m + 6 \cdot m_{\text{cla}})$  in SVGIC. We first observe that for each satisfied clause  $C_j$ , either  $c_{jt}$  is co-displayed to  $u_j$  and  $v_{jt}$ , or  $c'_{jt}$  is co-displayed to  $u_j$  and  $v'_{jt}$ ; each of the above cases achieves an objective value of 2. As there are at least  $m$  satisfied clauses, the above two cases in total contribute at least  $2 \cdot m$  to the objective of SVGIC. Next, for each Boolean variable  $a_i$  assigned **TRUE**,  $w_i$  is co-displayed  $c'_i$  with every  $v'_{jt}$  such that  $1_{jt}$  is  $a_i$  or  $\bar{a}_i$ ; similarly, for each Boolean variable  $a_i$  assigned **FALSE**,  $w_i$  is co-displayed  $c_i$  with every  $v_{jt}$  such that  $1_{jt}$  is  $a_i$  or  $\bar{a}_i$ . The above two cases achieve an objective value of 2 for each co-display pairs. As there are exactly  $(3 \cdot m_{\text{cla}})$  such pairs of  $(w_i, v_{jt})$  or  $(w_i, v'_{jt})$ , they in total contributes  $(6 \cdot m_{\text{cla}})$  to the objective of SVGIC. Combining the above two parts, the objective value is at least  $(2 \cdot m + 6 \cdot m_{\text{cla}})$ .

We then prove the necessary condition. Assume that all possible truth assignments in  $\phi$  satisfy fewer than  $m'$  clauses. We then prove that the optimal solution in the SVGIC instance is strictly less than  $(2 \cdot m' + 6 \cdot m_{\text{cla}})$ . We prove it by contradiction. If there exists a feasible solution of SVGIC with an objective at least  $(2 \cdot m' + 6 \cdot m_{\text{cla}})$ , we prove that there exists at least one feasible solution of SVGIC such that the total social utility from co-displaying items to one vertex in  $V_2$  and another vertex in  $V_3$ , denoted by  $\tau(V_2, V_3)$ , is exactly  $6 \cdot m_{\text{cla}}$ . To prove the above argument, we first observe that the total number of edges between vertices in  $V_2$  and vertices in  $V_3$  is  $6 \cdot m_{\text{cla}}$ . Furthermore, these edges form exactly  $3 \cdot m_{\text{cla}}$  copies of  $P_3$  (where every  $P_3$  is formed by a  $w_i$  and two vertices  $v_{jt}$  and  $v'_{jt}$  such that  $1_{jt}$  is  $a_i$  or is  $\bar{a}_i$ ). Since the two edges in each  $P_3$  can only contribute to the final objective through different co-displayed items, these edges in total contribute at most  $6 \cdot m_{\text{cla}}$  to the final objective. Therefore,  $\tau(V_2, V_3) \leq 6 \cdot m_{\text{cla}}$ . Next, suppose a solution with  $\tau(V_2, V_3) < 6 \cdot m_{\text{cla}}$  is given. For an arbitrary literal  $1_{jt}$ , let  $a_i$  be the corresponding variable. We examine the following cases: 1)  $w_i$  is co-displayed  $c_i$  with  $v_{jt}$  or  $c'_i$  with  $v'_{jt}$ . In this case, a social utility of 2 is achieved. 2)  $w_i$  is not co-displayed any item with  $v_{jt}$  or  $v'_{jt}$ . As  $w_i$  does not contribute to the objective if it is displayed neither  $c_i$  nor  $c'_i$ , it is feasible to display  $c_i$  to  $w_i$  without decreasing the total objective. Next, it is also feasible to co-display  $c_i$  to both  $v_{jt}$  and  $v'_{jt}$  while not decreasing the total objective, since the most social utility that  $v_{jt}$  or  $v'_{jt}$  could contribute previously from being co-displayed  $c_{jt}$  (or  $c'_{jt}$ ) is also at most 2. By repeatedly applying the two cases, we conclude that there exists a solution where each literal  $1_{jt}$  contributes exactly 2

to  $\tau(V_2, V_3)$ , which implies  $\tau(V_2, V_3) = 6 \cdot m_{\text{cla}}$ .

Given a feasible solution with  $\tau(V_2, V_3) = 6 \cdot m_{\text{cla}}$ , we construct a truth assignment in  $\phi$  according to this solution as follows. Let every variable  $a_i$  be **TRUE** if and only if  $w_i$  is displayed  $c'_i$ ; otherwise  $a_i$  is **FALSE**. Note that this is consistent to the construction in the sufficient condition. For each pair of vertices  $(u_j, v_{jt})$  co-displayed  $c_{jt}$  in the solution, since  $v_{jt}$  is not co-displayed an item with any vertex in  $V_3$ , it follows that  $v'_{jt}$  must be co-displayed  $c'_i$  with some  $w_i \in V_3$  (otherwise  $\tau(V_2, V_3) = 6 \cdot m_{\text{cla}}$  does not hold), which in turn implies  $a_i$  (appearing in clause  $C_j$ ) is assigned **TRUE** in this truth assignment, and  $C_j$  is therefore satisfied. Analogously, for  $(u_j, v'_{jt})$  co-displayed  $c'_{jt}$  in the solution,  $C_j$  is also satisfied by this truth assignment. As the total objective is at least  $(2 \cdot m' + 6 \cdot m_{\text{cla}})$  and  $\tau(V_2, V_3) = 6 \cdot m_{\text{cla}}$ , the total social utility from co-displaying items to one vertex in  $V_1$  and another vertex in  $V_2$ , denoted by  $\tau(V_1, V_2)$ , is at least  $(2 \cdot m')$ , meaning that at least  $m'$  clauses are satisfied in  $\phi$  by the constructed truth assignment, leading to a contradiction. Therefore, we conclude that the optimal solution in the SVGIC instance is strictly less than  $(2 \cdot m' + 6 \cdot m_{\text{cla}})$ .

Finally, for any instance of **MAX – E3SAT**, a folklore random assignment of variables satisfies  $\frac{7}{8} \cdot m_{\text{cla}}$  clauses in expectation (see, for example, Theorem 2.15 in [30]). Therefore, for any instance of **MAX – E3SAT** with  $m_{\text{cla}}$  clauses, the optimal objective is always at least  $\frac{7}{8} \cdot m_{\text{cla}}$ . Therefore, let  $\text{OPT}_{\text{E3SAT}}$  denote the optimal objective in **MAX – E3SAT**, we have

$$\frac{7}{8} \cdot m_{\text{cla}} \leq \text{OPT}_{\text{E3SAT}} \leq m_{\text{cla}}.$$

Let  $\chi$  be a constant. Based on this lower bound, we have:

$$\begin{aligned} \text{OPT}_{\text{E3SAT}} \geq \chi &\implies \text{OPT}_{\text{SVGIC}} \geq 2\chi + 6m_{\text{cla}} \\ &\geq 8\chi \\ \text{OPT}_{\text{E3SAT}} < \frac{7}{8} \cdot \chi &\implies \text{OPT}_{\text{SVGIC}} < 2 \cdot \frac{7}{8} \cdot \chi + 6m_{\text{cla}} \\ &< \frac{31}{4} \cdot \chi, \end{aligned}$$

where the second inequality holds since  $\frac{7}{8} \cdot m_{\text{cla}} \leq \text{OPT}_{\text{E3SAT}} < \frac{7}{8} \cdot \chi$  implies  $m_{\text{cla}} < \chi$ . The lemma follows.  $\square$

Therefore, combining Lemma 1 and 2, if there exists an algorithm  $\mathcal{ALG}$  that approximates SVGIC within a ratio of  $\frac{8}{31} = \frac{32}{31}$  (or equivalently, solves the  $\frac{32}{31}$ -gap version of SVGIC), then it also solves the  $\frac{8}{7}$ -gap version of **MAX – E3SAT**, which is known to be NP-hard. This completes the proof of Theorem 2.

We then prove the APX-hardness through an exact reduction from the maximum  $K_3$ -packing problem (**Max-K3P**) [10] as follows.

*Proof.* 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  $\Delta$  consisting of vertices  $u, v$ , and  $w$  in  $\hat{G}$ , we construct an item  $c_\Delta$ . Let social utility  $\tau(u, v, c_\Delta) = \tau(v, u, c_\Delta) = \tau(u, w, c_\Delta) = \tau(w, u, c_\Delta) = \tau(w, v, c_\Delta) = \tau(v, w, c_\Delta) = 0.5$ .

$\tau(w, u, c_\Delta) = \tau(v, w, c_\Delta) = \tau(w, v, c_\Delta) = 0.5$ . 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  $\Delta$  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  $c_\Delta$ ). 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 [10], SVGIC is also APX-hard.  $\square$

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) [55]. 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  $S = \{w_1, w_2, \dots, w_{|S|}\}$  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  $\mathcal{C} = \{c_1, c_2, \dots, c_m\}$ , 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 [55], 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.  $\square$

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. We revisit this issue in Section 3.4.

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  $x_{u,s}^c$  denote whether user  $u$  is displayed item  $c$  at slot  $s$ , i.e.,  $x_{u,s}^c = 1$  if and only if  $\mathbf{A}(u, s) = c$ . Let  $x_u^c$  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  $y_{e,s}^c$  denote whether  $u$  and  $v$  are co-displayed item  $c$  at slot  $s$ , i.e.,  $y_{e,s}^c = 1$  if and only if  $u \xleftrightarrow[s]{} v$ . Similarly, variable  $y_e^c = 1$  if and only if  $u \xleftrightarrow{} v$ . The objective of SVGIC is specified as follows.

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

subject to the following constraints,

$$\sum_{s=1}^k x_{u,s}^c \leq 1, \quad \forall u \in V, c \in \mathcal{C} \quad (1)$$

$$\sum_{c \in \mathcal{C}} x_{u,s}^c = 1, \quad \forall u \in V, s \in [k] \quad (2)$$

$$x_u^c = \sum_{s=1}^k x_{u,s}^c, \quad \forall u \in V, c \in \mathcal{C} \quad (3)$$

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

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

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

$$x_{u,s}^c, x_u^c, y_{e,s}^c, y_e^c \in \{0, 1\}, \quad \forall u \in V, e \in E, s \in [k], c \in \mathcal{C}. \quad (7)$$

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  $x_u^c = 1$  ( $u$  is displayed  $c$  in the configuration) if and only if there is exactly one slot  $s$  with  $x_{u,s}^c = 1$ . Similarly, constraint (4) ensures  $y_e^c = 1$  if and only if  $y_{e,s}^c = 1$  for exactly one  $s$ . Constraints (5) and (6) specify the co-display, where  $y_{e,s}^c$  is allowed to be 1 only if  $c$  is displayed to both  $u$  and  $v$  at slot  $s$ , i.e.,  $x_{u,s}^c = x_{v,s}^c = 1$ . Finally, constraint (7) ensures all decision variables are binary. Note that the  $x$  variables in the IP model are sufficient to represent the solution of SVGIC (i.e.,  $x_{u,s}^c$  denotes whether an item  $c$  is displayed at slot  $s$  for user  $u$ ), whereas the  $y$  variables are auxiliary to enable SVGIC to be formulated as an IP. Without incorporating the  $y$  variables, the formulation will become nonlinear.

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

$$\max \sum_{u \in V} \sum_{c \in \mathcal{C}} [(1 - \lambda) \cdot p(u, c) \cdot x_u^c + \lambda \cdot \sum_{e=(u,v) \in E} \tau(u, v, c) \\ \cdot ((1 - d_{tel}) \cdot y_e^c + d_{tel} \cdot z_e^c)]$$

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

$$z_e^c \leq x_u^c, \quad \forall e = (u, v) \in E, c \in \mathcal{C} \quad (8)$$

$$z_e^c \leq x_v^c, \quad \forall e = (u, v) \in E, c \in \mathcal{C} \quad (9)$$

$$x_{u,s}^c, x_u^c, y_{e,s}^c, y_e^c, z_e^c \in \{0, 1\}, \quad \forall u \in V, e \in E, s \in [k], c \in \mathcal{C}. \quad (10)$$

Constraints 8 and 9 specify the co-display, where  $z_e^c$  is allowed to be 1 only if  $c$  is displayed to  $u$  and  $v$  in some (not necessarily the same) slots, i.e.,  $x_u^c = x_v^c = 1$ . Therefore,  $y_e^c = z_e^c = 1$  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  $y_e^c = 1$  for some edge  $e$ , the corresponding  $z_e^c$  will also naturally equal 1, as it is always better to have a larger  $z_e^c$ . Therefore, no additional constraint needs to be introduced to enforce this relation. Note that, similar to that in the IP for SVGIC, the  $y$  and  $z$  variables are auxiliary variables to enable linearity in this IP.

### 3.4 Related Combinatorial Problems

SVGIC is related to the *Multi-Labeling* (ML) [88] problem and its variations, including Multiway Partition [88, 96], *Maximum Happy Vertices/Edges* (MHV/MHE) [92], and *Multiway Cut* [16] in graphs. In the Multi-Labeling (ML) problem, the general inputs are a ground set of entities  $V$ , a set of labels  $L = \{1, 2, \dots, m\}$ , and a *partial* labeling function  $\ell : V \mapsto L$  that pre-assigns each label  $i$  to a non-empty subset  $T_i \subseteq V$ , and a set function  $f : 2^V \mapsto \mathcal{R}$ . The goal of ML is to partition the ground set into  $m$  subsets to optimize the aggregated set function on the partitioned subsets, i.e.,  $\sum_{i=1}^m f(T_i)$ . The special cases with set function  $f$  being submodular or supermodular is called *Submodular Multi-Labeling* (Sub-ML) or *Supermodular Multi-Labeling* (Sup-ML) [88]. Another work [96] studies the case where the partial labeling function is required to assign a label  $i$  to only one entity  $v_i$ . The above problems become *Submodular Multiway Partition* (Sub-MP) and *Supermodular Multiway Partition* (Sup-MP) in this context. Some special cases of Sub-ML and Sup-ML are described as graph problems, where the ground set  $V$  is the vertex set on a graph  $G = (V, E)$ . In this context, the above assumption on  $f$  in Sub-MP and Sup-MP can be interpreted as  $m$  specified terminals on the graph. Two representative special cases are the *Maximum Happy Vertices* (MHV) problem and the *Maximum Happy Edges* (MHE) problem [92]. Originally proposed to study homophily effects in networks, MHV and MHE are described as coloring problems where, different from traditional graph coloring, vertices are encouraged to share the same color with neighbors. *Edges with same-color vertices* and *vertices sharing the same color with all neighbors* are characterized as *happy edges* and *happy vertices*, respectively. The optimization goals of MHV and MHE are thus to maximize the number of happy vertices and edges. The complement problem of MHV, *Minimum Unhappy Vertices* (MUHV) [88], minimizes the number of unhappy vertices. When the above optimization goal of MHE is combined with the assumption

of  $m$  terminals, the resultant problem becomes the *Multiway Uncut* problem [43], which is the complement of the classical and extensively studied *Multiway Cut* problem [16]. The Multiway Cut problem asks for the minimum number of edges removed to partition a graph into  $m$  subgraph, each containing one terminal. The Metric Labeling problem [41] is a generalization of Multiway Cut considering edge weights and also a cost function on labeling the vertices. We summarize the problem characteristics, hardness results, currently best algorithmic results, and their interdependency in Table 4. All the problems are NP-hard, while more advanced inapproximability results, relying on other complexity conjectures, are stated for some problems. Note that the approximation ratios for maximization problems are shown as inverses (i.e., values larger than 1) for better comparison and consistency with this paper.

Among the ML-type problems, SVGIC is particularly correlated to the MHE problem. Regarding the displayed items in SVGIC as colors(labels) in MHE, the social utility achieved in SVGIC is closely related to preserved number of happy edges in MHE, and SVGIC thereby encourages partitioning all users into dense subgroups to preserve the most social utility. However, SVGIC is more difficult than the above ML-type problems due to the following reasons. 1) The ML-type problems find a strict partition that maps each entity/vertex to only one color(label), while SVGIC assigns  $k$  items to each user, implying that any direct reduction from the above problems can only map to the  $k = 1$  special case in SVGIC, where AVG can achieve a 2-approximation. 2) The above problems do not discriminate between different labels, i.e., switching the assigned labels of two different subgroups will not change their objective functions. This corresponds to the special case of SVGIC where all preference utility  $p(u, c)$  and social utility  $\tau(u, v, c)$  do not depend on the item  $c$ . In contrast, SVGIC captures the item preferences as both utility measures in SVGIC depend on the item identity. 3) The above problems admit a partial labeling (pre-labeling) in the input such that some entities have predefined fixed labels (otherwise labeling every entity with the same label is optimal and renders the problem trivial), while SVGIC does not specify any items to be displayed to specific users. However, SVGIC requires  $k$  items for each user; moreover, even in the  $k = 1$  special case, it is still not optimal to simply display the same item to all users in SVGIC due to the item-dependent preference and social utility.

Finally, we note that while most of the minimization problems admit stronger hardness results than conventional NP-hardness, no such type of results are known for all maximization problems above. As SVGIC is also a maximization problem, its hardness does not directly follow from any complementary minimization problems. Interestingly, the algorithmic aspect of all these problems share an obvious trend: almost all approximation results here, including our algorithm for SVGIC, utilizes some kind of (dependent) randomized rounding on some formulation of relaxed problem (not limited to linear relaxation). In this regard, this paper extends the scheme beyond labeling problems to the SVGIC problem that assigns a  $k$ -itemset to each user, which has an even more complicated combinatorial nature.

## 4. ALGORITHM DESIGN

**Table 4: Related Combinatorial Problems (approximation ratio for MAX problems inversed).**

Problem Name	Abbrev.	Prelabeling	Type	Objective	Hardness Result	Approx. Result	Is a special case of	Is the complement of
Submodular Multi-Labeling [88]	Sub-ML	General	MIN	submodular set function	$2 - \frac{2}{m} - \epsilon$ (UGC) [88]	$2 - \frac{2}{m}$ [88]		Sup-ML
Supermodular Multi-Labeling [88]	Sup-ML	General	MAX	supermodular set function	NP-hard	$\frac{m}{2}$ [88]		Sub-ML
Submodular Multiway Partition [96]	Sub-MP	$m$ terminals	MIN	submodular set function	$2 - \frac{2}{m} - \epsilon$ (NP = RP) [19]	$2 - \frac{2}{m}$ [19]	Sub-ML	Sup-MP
Supermodular Multiway Partition [88]	Sup-MP	$m$ terminals	MAX	supermodular set function	NP-hard		Sup-ML	Sub-MP
Maximum Happy Vertices [92]	MHV	General	MAX	# happy vertices	NP-hard	$\frac{m}{2}$ [88]	Sup-ML	MUHV
Minimum Unhappy Vertices [88]	MUHV	General	MIN	# unhappy vertices	$2 - \frac{2}{m} - \epsilon$ (UGC) [88]	$2 - \frac{2}{m}$ [88]	Sub-ML	MHV
Maximum Happy Edges [92]	MHE	General	MAX	# happy edges	NP-hard	1.1716 [93]	Sup-ML	
Multiway Cut [16]		$m$ terminals	MIN	# edges removed	$\frac{8}{7}$ [54]	1.2965 [68]	UML	Multiway Uncut
Multiway Uncut [43]		$m$ terminals	MAX	# edges preserved	NP-hard	1.1716 [43]	MHE	Multiway Cut
Uniform Metric Labeling [41]	UML	$m$ terminals	MIN	# total cost of edges/vertices	$2 - \frac{2}{m}$ (UGC) [54]	2 [41]	Metric Labeling	

**Table 5: Notations used in AVG.**

Symbol	Description
$X^*$	optimal fractional solution
$x_{u,s}^{*c}$	optimal decision variable
$(c, s, \alpha)$	set of focal parameters
$c/\hat{c}$	focal item
$s/\hat{s}$	focal slot
$u/\hat{u}$	current user/eligible user
$\alpha$	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
$S_{\text{cur}}$	available display units
$S_{\text{tar}}(c, s, x_{u,s}^{*c})$	display units to be filled
$S_{\text{fut}}(c, s, x_{u,s}^{*c})$	display units not to be filled
$\text{ALG}(S_{\text{tar}}(c, s, x_{u,s}^{*c}))$	SAVG utility gained by CSF
$\text{OPT}_{\text{LP}}(S_{\text{fut}}(c, s, x_{u,s}^{*c}))$	expected future SAVG utility
$w_e^c$	$\tau(\hat{u}, \hat{v}, c) + \tau(\hat{v}, \hat{u}, c)$
$r$	balancing ratio

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 [85]

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 3.3 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  $x_{u,s}^{*c}$  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 [41], 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  $x_{u,s}^c$  greater than a *grouping threshold*  $\alpha$ . 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 [41], 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.

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.,  $p'(u,c) = \frac{1-\lambda}{\lambda} p(u,c)$ , 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 5.

## 4.1 LP Relaxation and an Independent Rounding Scheme

Following the standard linear relaxation technique [86], the LP relaxation of SVGIC is formulated by replacing the integrality constraint (constraint (7)) in the IP model, i.e.,  $x_{u,s}^c, x_u^c, y_{e,s}^c, y_e^c \in \{0,1\}$ , with linear upper/lower bound constraints, i.e.,  $0 \leq x_{u,s}^c, x_u^c, y_{e,s}^c, y_e^c \leq 1$ . The optimal fractional solution of the relaxed problem can be acquired in polynomial time with commercial solvers, e.g., CPLEX [37] or Gurobi [29]. Recall that the  $x$ -variables are sufficient to represent the solution of SVGIC (i.e.,  $x_{u,s}^c$  denotes whether an item  $c$  is displayed at slot  $s$  for user  $u$ ), whereas the  $y$ -variables are auxiliary. Therefore, the optimal solution can be fully represented by  $X^*$  (the set of optimal  $x$  variables). The fractional decision variable  $x_{u,s}^{*c}$  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.

**Table 6: Optimal fractional solution for slot 1 in Example 2.**

	$x_{\cdot,1}^{*c_1}$	$x_{\cdot,1}^{*c_2}$	$x_{\cdot,1}^{*c_3}$	$x_{\cdot,1}^{*c_4}$	$x_{\cdot,1}^{*c_5}$
$u_A$	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

**Example 3.** Table 6 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  $x_{u,1}^{*c_1} = x_{u,2}^{*c_1} = x_{u,3}^{*c_1} = 0.33$ .

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  $x_{u,s}^{*c}$ , 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**  
1: **for**  $u \in V$  **do**  
2:   **for**  $s \in \{1, 2, \dots, k\}$  **do**  
3:     Display item  $c$  to user  $u$  at slot  $s$  independently  
      with probability  $x_{u,s}^{*c}$

---

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 3.** 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  $p(u, c) = 0$  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  $x_{u,s}^{*c} = \frac{1}{m}$  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.  $\square$

## 4.2 Alignment-aware Algorithm

To address the above issues, we devise the *Co-display Subgroup Formation* (CSF) rounding scheme, inspired by the dependent rounding scheme for labeling problems [41], 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  $(c, s, \alpha)$  with  $c \in \mathcal{C}, s \in \{1, 2, \dots, k\}$ , and  $\alpha \in [0, 1]$  uniformly at random; it then 2) conducts CSF according to the selected set of parameters  $(c, s, \alpha)$  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  $\mathbf{A}$

- 1:  $\mathbf{A}(\hat{u}, \hat{s}) \leftarrow \text{NULL}$  for all  $\hat{u}, \hat{s}$
- 2:  $X^* \leftarrow X_{\text{LP}}^*$
- 3: **while** some entry in  $\mathbf{A}$  is NULL **do**
- 4:     Sample  $c \in \mathcal{C}, s \in [k], \alpha \in [0, 1]$  randomly
- 5:     **for**  $\hat{u} \in V$  **do**
- 6:         **if**  $\mathbf{A}(\hat{u}, s) = \text{NULL}$  and  $\mathbf{A}(\hat{u}, t) \neq c \forall t \neq s$  **then**
- 7:             ( $\hat{u}$  eligible for  $(c, s)$ )
- 8:         **if**  $x_{\hat{u}, s}^* \geq \alpha$  **then**
- 9:              $\mathbf{A}(\hat{u}, s) \leftarrow c$
- 10: **return**  $\mathbf{A}$

---

**Co-display Subgroup Formation.** Given the randomly sampled set of parameters  $(c, s, \alpha)$ , 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  $x_{\hat{u}, s}^* \geq \alpha$ . In other words, given  $(c, s, \alpha)$ , CSF co-displays  $c$  to a target subgroup  $U$  that consists of every eligible user  $\hat{u}$  with  $x_{\hat{u}, s}^* \geq \alpha$ . Therefore, the grouping threshold  $\alpha$  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  $c$ ,  $\Pr(u \leftrightarrow v) \geq \frac{y_e^c}{4}$ ; 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 3 where independent rounding performs poorly. By exploiting the

dependent rounding scheme in CSF, since  $x_{u, s}^{*c} = \frac{1}{m}$  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 6, assume that the set of focal parameters are sampled as  $(c, s, \alpha) = (c_1, 3, 0.06)$ . Since  $x_{u_A, 3}^{*c_1} = x_{u_B, 3}^{*c_1} = x_{u_D, 3}^{*c_1} = 0.33 > 0.06 > x_{u_C, 3}^{*c_1} = 0$ , 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  $x_{u_B, 2}^{*c_4} = x_{u_C, 2}^{*c_4} = x_{u_D, 2}^{*c_4} = 0.33 > 0.22 > x_{u_A, 2}^{*c_4} = 0$ . In the third iteration, RFS selects  $(c, s, \alpha) = (c_3, 1, 0.04)$ . As only  $x_{u_C, 1}^{*c_3} = 0.33$  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  $x_{u_C, 5}^{*c_3} = 0.33 > 0.2$ , 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  $x_{u_C, 5}^{*c_3}$  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 7, achieving a total SAVG utility of 9.75.<sup>1</sup>  $\square$

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  $(c, s, \alpha)$  and 2) CSF with  $(c, s, \alpha)$ . 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  $x_{u,s}^c, x_u^c, y_{e,s}^c$ , and  $y_e^c$  in Section 3.3. Let  $w_e^c = \tau(u, v, c) + \tau(v, u, c)$  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

$$\begin{aligned}\mathbb{E}[\mathcal{R}_{\text{pre}}] &= \mathbb{E}[\sum_{c \in \mathcal{C}} \sum_{u \in V} p(u, c) \cdot x_u^c] \\ \mathbb{E}[\mathcal{R}_{\text{soc}}] &= \mathbb{E}[\sum_{c \in \mathcal{C}} \sum_{e \in E} w_e^c \cdot y_e^c]\end{aligned}$$

<sup>1</sup>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 Theorem 4 and also in the enhancement strategies in Section 4.4.

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

**Lemma 4.** In any iteration  $t$ , if  $u$  is eligible for  $(c, s)$ , the probability  $P_{\text{rec}}^u$  that  $\mathbf{A}(u, s) \leftarrow c$  is  $\frac{x^*_{u,s}}{k \cdot m}$  (where  $k$  and  $m$  are respectively the numbers of slots and items) since  $c$  and  $s$  are selected randomly, and  $\alpha$  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 5.** In any iteration  $t$ , for any  $c, s$  and a user  $u$  eligible for  $(c, s)$ , the probability  $P_{\text{ne}}^u$  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 4, the probabilities for the above two cases are at most  $\sum_{\hat{s}=1}^k \frac{x^*_{u,\hat{s}}}{k \cdot m}$  and  $\sum_{\hat{c} \in C} \frac{x^*_{u,s}}{k \cdot m}$ , respectively.

Recall that for any  $u$ ,  $\sum_{\hat{s}=1}^k x^*_{u,\hat{s}} \leq 1$  and  $\sum_{\hat{c} \in C} x^*_{u,s} = 1$  in LP relaxation. Therefore, the total probability of the above cases is at most  $\frac{1}{k \cdot m} (\sum_{\hat{s}=1}^k x^*_{u,\hat{s}} + \sum_{\hat{c} \in C} x^*_{u,s}) \leq \frac{2}{k \cdot m}$ .  $\square$

In the following, we first consider the case that  $u$  is eligible for  $(c, s)$  in the beginning of iteration  $t$ . According to Lemma 4, the probability  $P_{\text{rec}}$  that  $u$  is displayed  $c$  in slot  $s$  in this iteration is  $P_{\text{rec}} = \frac{x^*_{u,s}}{k \cdot m}$ . Moreover, according to above, let  $P_{\text{ne}}$  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

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

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,  $\mathbb{E}[\mathcal{R}_{\text{per}}] \geq \sum_{c \in C} \sum_{u \in V} (\text{p}(u, c) \cdot \frac{x^*_{u,s}}{2})$ . Next, we aim to use a similar approach to prove that  $\mathbb{E}[\mathcal{R}_{\text{soc}}] \geq \frac{1}{4} \sum_{c \in C} \sum_{e \in E} w_e^c \cdot y_e^c$ . 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 6.** 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  $\frac{\max\{x^*_{u,s}, x^*_{v,s}\}}{k \cdot m}$ , and the probability  $P_{\text{rec}}^e$  that  $\mathbf{A}(u, s) \leftarrow c$  and  $\mathbf{A}(v, s) \leftarrow c$  is  $\frac{\min\{x^*_{u,s}, x^*_{v,s}\}}{k \cdot m}$ .

The reason of Observation 6 is as follows. If  $\alpha \leq \min_{u \in U} x^*_{u,s}$ , CSF assigns  $c$  at  $s$  to all users in  $U$  since  $x^*_{u,s} \geq \alpha$  for all  $u \in U$ . Similarly, if  $\alpha \leq \max_{u \in U} x^*_{u,s}$ , CSF at least assigns  $c$  to the user  $u$  with the largest  $x^*_{u,s}$ . 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 7.** In any iteration  $t$ , for any pair of users  $e = (u, v)$  with  $u$  and  $v$  eligible for  $(c, s)$ , the probability  $P_{\text{ne}}^e$  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 6, the probabilities for the above two cases are at most  $\sum_{\hat{s}=1}^k \frac{\max\{x^*_{u,\hat{s}}, x^*_{v,\hat{s}}\}}{k \cdot m}$  and  $\sum_{\hat{c} \in C} \frac{\max\{x^*_{u,s}, x^*_{v,s}\}}{k \cdot m}$ , respectively. Recall that for any  $u$ ,  $\sum_{\hat{s}=1}^k x^*_{u,\hat{s}} \leq 1$  and  $\sum_{\hat{c} \in C} x^*_{u,s} = 1$  in LP relaxation. Therefore, the total probability of the above cases is at most

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

$\square$

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

we have

$$\begin{aligned}
\Pr[u \xrightarrow[s]{c} v] &= \sum_{t=1}^{\infty} \Pr[u \xrightarrow[s]{c} v|_t] \\
&= \sum_{t=1}^{\infty} P_{\text{rec}}^e \cdot \Pr[u, v \text{ are both eligible for } (c, s) \text{ in } t\text{-th iteration}] \\
&= \sum_{t=1}^{\infty} P_{\text{rec}}^e \cdot \Pr[u, v \text{ both eligible for } (c, s) \text{ in } t\text{-th iteration}] \\
&= \sum_{t=1}^{\infty} P_{\text{rec}}^e \cdot (1 - P_{\text{ne}}^e)^{t-1} = \frac{P_{\text{rec}}^e}{P_{\text{ne}}^e} \\
&\geq \frac{\frac{\min\{x_{u,s}^*, x_{v,s}^*\}}{k \cdot m}}{\frac{4}{k \cdot m}} \\
&\geq \frac{\min\{x_{u,s}^*, x_{v,s}^*\}}{4} = \frac{y_{e,s}^*}{4}.
\end{aligned}$$

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

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

Therefore,

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

which proves the approximation ratio.  $\square$

In the above derivation, a large  $\alpha$  could lead to an empty target group if it exceeds  $x_{u,s}^*$  for every  $u$  eligible for  $(c, s)$ . Therefore, the total number of iterations could approach  $\infty$ . To address the above issue, instead of sampling  $(c, s, \alpha)$  uniformly from all possible combinations, AVG samples  $(c, s, \alpha)$  uniformly from only the combinations generating nonempty target groups (i.e., an enormous  $\alpha$  is no longer chosen). Because the setting of  $\alpha$  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<sup>2</sup> of solving  $X^*$ . Some 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  $\beta$ -approximation of the LP relaxation problem, AVG returns an expected  $(4 \cdot \beta)$ -approximate SAVG  $k$ -Configuration.

**Corollary 4.3.** For  $k = 1$ , given the optimal fractional solution  $X^*$ , AVG returns an expected 2-approximate SAVG  $k$ -Configuration in  $O(n^2 \cdot k)$ -time.

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

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

Therefore, if AVG is repeated  $n_{\text{repeat}}$  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.

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

Finally, for the third corollary, simply note that for  $k = 1$ , the final sum in Lemma 7 changes from  $\frac{4}{k \cdot m}$  to  $\frac{2}{k \cdot m}$ , since the first case in the proof of Lemma 7 is impossible to happen in  $k = 1$ . Following the original proof but plugging in this change implies the 2-approximation of AVG when  $k = 1$ .  $\square$

The second corollary is particularly useful in practice for improving the efficiency of AVG since state-of-the-art LP

<sup>2</sup>The current best time complexity for solving linear program equals the complexity of matrix multiplication, or roughly  $O(N^{2.5})$  for  $N$  variables [15]. However, practical computation generally takes much less time.

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  $\alpha$  plays a key role in forming effective target subgroups in CSF. If  $\alpha$  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  $\alpha$  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  $\alpha$  can be assigned in a discrete manner.

**Observation 1.** There are  $O(knm)$  distinct possible outcomes in CSF, each corresponding to a grouping threshold  $\alpha = x_{u,s}^c$ , 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  $\alpha$  set to the smallest  $x_{u,s}^c \geq x$ . It enables us to derandomize AVG effectively. Instead of randomly sampling  $(c, s, \alpha)$ , we carefully evaluate the outcomes (of CSF) from setting  $\alpha$  to every possible  $x_{u,s}^c$  in the optimal fractional solution. Intuitively, it is desirable to select an  $\alpha$  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  $S_{\text{tar}}(c, s, x_{u,s}^c) = \{(\hat{u}, \hat{s}) \in S_{\text{cur}} | x_{\hat{u}, \hat{s}}^c \geq x_{u,s}^c\}$  denote the set of display units to be assigned  $c$  in CSF with  $\alpha = x_{u,s}^c$ . Let  $S_{\text{fut}}(c, s, x_{u,s}^c) = S_{\text{cur}} \setminus S_{\text{tar}}(c, s, x_{u,s}^c)$  denote the set of remaining display units to be processed in the future. Moreover, let  $\text{ALG}(S_{\text{tar}}(c, s, x_{u,s}^c))$  be the SAVG utility gained by co-displaying  $c$  to the target subgroup. Let  $\text{OPT}_{\text{LP}}(S_{\text{fut}}(c, s, x_{u,s}^c))$  represent the expected SAVG utility acquired in the future from the display units in  $S_{\text{fut}}(c, s, x_{u,s}^c)$ . To strike a balance between the increment of SAVG utility in the current iteration and in the future, AVG-D examines every  $\alpha = x_{u,s}^c$  to maximize

$$f(c, s, x_{u,s}^c) = \text{ALG}(S_{\text{tar}}(c, s, x_{u,s}^c)) + r \cdot \text{OPT}_{\text{LP}}(S_{\text{fut}}(c, s, x_{u,s}^c)),$$

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

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

**Table 8: SAVG  $k$ -Configuration returned by AVG-D for Example 4.**

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

where

$$\begin{aligned} \text{ALG}(S_{\text{tar}}(c, s, x_{u,s}^c)) &= \sum_{(\hat{u}, \hat{s}) \in S_{\text{tar}}} p(\hat{u}, c) + \sum_{\substack{e=(\hat{u}, \hat{v}) \\ (\hat{u}, \hat{s}), (\hat{v}, s) \in S_{\text{tar}}}} w_e^c, \\ \text{OPT}_{\text{LP}}(S_{\text{fut}}(c, s, x_{u,s}^c)) &= \sum_{c \in \mathcal{C}} \sum_{s=1}^k \sum_{(\hat{u}, \hat{s}) \in S_{\text{fut}}} p(\hat{u}, c) x_{\hat{u}, \hat{s}}^c \\ &\quad + \sum_{\substack{e=(\hat{u}, \hat{v}) \\ (\hat{u}, \hat{s}), (\hat{v}, s) \in S_{\text{fut}}}} w_e y_{e, s}^{*c}, \end{aligned}$$

$w_e^c = \tau(\hat{u}, \hat{v}, c) + \tau(\hat{v}, \hat{u}, c)$  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. It is summarized in Algorithm 3.

**Algorithm 3** Deterministic Alignment-aware VR Subgroup Formation

---

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

- 1:  $\mathbf{A}(\hat{u}, \hat{s}) \leftarrow \text{NULL}$  for all  $\hat{u}, \hat{s}$
- 2:  $X^* \leftarrow X_{\text{LP}}^*$
- 3: **while** some entry in  $\mathbf{A}$  is **NULL** **do**
- 4:      $c, s, u \leftarrow \arg \max_{c, s, u} f(c, s, x_{u,s}^c)$
- 5:      $\alpha \leftarrow x_{u,s}^c$
- 6:     **for**  $\hat{u} \in V$  **do**
- 7:         **if**  $\mathbf{A}(\hat{u}, s) = \text{NULL}$  and  $\mathbf{A}(\hat{u}, t) \neq c \forall t \neq s$  **then**
- 8:             **if**  $x_{\hat{u}, s}^c \geq \alpha$  **then**
- 9:                  $\mathbf{A}(\hat{u}, s) \leftarrow c$
- 10: **return**  $\mathbf{A}$

---

**Example 5.** For the instance in Example 2, the first iteration of AVG-D with  $r = \frac{1}{4}$  sets  $\alpha = x_{u_B, 1}^{*c_5} = 0$ , and CSF with  $(c, s, x_{u,s}^c) = (c_5, 1, 0)$  co-displays the SP camera to everyone at slot 1, where  $S_{\text{tar}}(c, s, x_{u,s}^c) = \{(u_A, 1), (u_B, 1), (u_C, 1), (u_D, 1)\}$ ,  $S_{\text{fut}}(c, s, x_{u,s}^c) = \{(u_A, 2), (u_B, 2), (u_C, 2), (u_D, 2), (u_A, 3), (u_B, 3), (u_C, 3), (u_D, 3)\}$ ,  $\text{ALG}(S_{\text{tar}}(c, s, x_{u,s}^c)) = p(u_A, c_5) + p(u_B, c_5) + p(u_C, c_5) + p(u_D, c_5) + w_{(u_A, u_B)}^{c_5} + w_{(u_A, u_C)}^{c_4} + w_{(u_A, u_D)}^{c_4} + w_{(u_B, u_C)}^{c_4} = 3.35$ ,  $\text{OPT}_{\text{LP}}(S_{\text{fut}}(c, s, x_{u,s}^c)) = 6.97$ , and  $f(c, s, x_{u,s}^c) = 3.35 + \frac{1}{4} \cdot 6.97 = 5.09$  is maximized. The next iteration selects  $\alpha = x_{u_A, 2}^{*c_1} = 0.33$  to co-display the tripod to  $\{\text{Alice, Bob, Dave}\}$  at slot 2. AVG-D selects

**Table 9: 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			
$\{u_A, u_B\}$	$c_2$	$c_1$	$c_5$
$\{u_C, u_D\}$	$c_4$	$c_5$	$c_3$

$\alpha = x^{*c_2}_{u_D,3} = 0$ ,  $\alpha = x^{*c_4}_{u_A,3} = 0$ , 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 8.

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  $\sum_{u \in V} p(u, c_1) + \sum_{u, u' \in V} \tau(u, u', c_1)$ , 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 9. 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.  $\square$

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. We also empirically evaluate the performance of AVG-D with different values of  $r$  in Section 6.7.

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

$$\begin{aligned} \text{OPT}_{\text{LP}}(S_{\text{cur}}) &= \sum_{c \in \mathcal{C}} \sum_{s=1}^k \sum_{\substack{(\hat{u}, \hat{s}) \in \\ S_{\text{cur}}(c, s, x^{*c}_{u, s})}} (p(\hat{u}, \hat{c}) x^{*\hat{c}}_{\hat{u}, \hat{s}} \\ &\quad + \sum_{\substack{e=(\hat{u}, \hat{v}) \\ (\hat{v}, \hat{s}) \in S_{\text{cur}}(c, s, x^{*c}_{u, s})}} w_e^{\hat{c}} y^{*\hat{c}}_{e, \hat{s}}). \end{aligned}$$

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  $(c, s, \alpha)$  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  $\frac{\text{OPT}_{\text{LP}}(S_{\text{cur}})}{4}$ . Thus, for each iteration there exists at least one  $(c, u, s)$  with  $f(c, s, x^{*c}_{u, s}) \geq \frac{\text{OPT}_{\text{LP}}(S_{\text{cur}})}{4}$ , and AVG-D is therefore a 4-approximating algorithm.

We begin with proving that in every iteration  $t$ , if  $(c, s, \alpha)$  is selected with RPS instead of DPS,  $\mathbb{E}[f(c, s, \alpha)] \geq \frac{1}{4} \cdot \text{OPT}_{\text{LP}}(S_{\text{cur}})$ , 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  $\frac{1}{4} \cdot \text{OPT}_{\text{LP}}(S_{\text{cur}})$ , since DPS maximizes  $f(c, s, x^{*c}_{u, s})$ .

**Lemma 8.** In any iteration of AVG, if  $(c, s, \alpha)$  is selected by RPS,  $\mathbb{E}[f(c, s, \alpha)] \geq \frac{1}{4} \cdot \text{OPT}_{\text{LP}}(S_{\text{cur}})$ .

*Proof.* In any iteration  $t$ , for each  $e = (\hat{u}, \hat{v})$ ,  $c$ , and  $s$  with  $(\hat{u}, s)$  and  $(\hat{v}, s)$  in  $S_{\text{cur}}$  (i.e.,  $\hat{u}$  and  $\hat{v}$  are eligible for  $(c, s)$ ), by Lemma 6, 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,

$$\begin{aligned} &\mathbb{E}[\text{ALG}(S_{\text{tar}}(c, s, \alpha))] \\ &= \mathbb{E}\left[\sum_{(\hat{u}, s) \in S_{\text{tar}}(c, s, \alpha)} (p(\hat{u}, c) + \sum_{(\hat{v}, s) \in S_{\text{tar}}(c, s, \alpha)} \tau(\hat{u}, \hat{v}, c))\right] \\ &= \sum_{c \in \mathcal{C}} \sum_{s=1}^k \left( \sum_{(\hat{u}, s) \in S_{\text{cur}}} p(\hat{u}, c) \frac{x^{*c}_{\hat{u}, s}}{k \cdot m} + \sum_{(\hat{u}, s), (\hat{v}, s) \in S_{\text{cur}}} w_e^c \frac{y^{*c}_{e, s}}{k \cdot m} \right) \\ &= \frac{1}{k \cdot m} \sum_{c \in \mathcal{C}} \sum_{s=1}^k \left( \sum_{(\hat{u}, s) \in S_{\text{cur}}} p(\hat{u}, c) x^{*c}_{\hat{u}, s} + \sum_{\substack{e=(\hat{u}, \hat{v}) \\ (\hat{v}, s) \in S_{\text{cur}}}} w_e^c y^{*c}_{e, s} \right) \\ &= \frac{\text{OPT}_{\text{LP}}(S_{\text{cur}})}{k \cdot m}. \end{aligned}$$

Next, from Lemma 5 and 7, for each  $e = (\hat{u}, \hat{v})$ ,  $\hat{c}$ , and  $\hat{s}$  such that  $(\hat{u}, \hat{s}), (\hat{v}, \hat{s}) \in S_{\text{cur}}$ , the probability that  $(\hat{u}, \hat{s}) \in S_{\text{fut}}(c, s, \alpha)$  is at least  $1 - P_{\text{ne}}^{\hat{u}}$ , and the probability that  $(\hat{u}, \hat{s}), (\hat{v}, \hat{s}) \in S_{\text{fut}}(c, s, \alpha)$  is at least  $1 - P_{\text{ne}}^{\hat{v}}$ , 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  $w_e^{\hat{c}} \cdot (1 - \frac{4}{k \cdot m})$ . Thus,

$$\begin{aligned} & \mathbb{E}[\text{OPT}_{\text{LP}}(S_{\text{fut}}(c, s, \alpha))] \\ & \geq \sum_{\hat{c} \in \mathcal{C}} \sum_{\hat{s}=1}^k \left[ \sum_{(\hat{u}, \hat{s}) \in S_{\text{cur}}} (1 - P_{\text{ne}}^{\hat{u}}) p_{\hat{u}, \hat{s}}^{\hat{c}} x^{*\hat{c}}_{\hat{u}, \hat{s}} + \sum_{\substack{(\hat{u}, \hat{s)}, \\ (\hat{v}, \hat{s}) \in S_{\text{cur}}}} (1 - P_{\text{ne}}^{\hat{v}}) w_e^{\hat{c}} y^{*\hat{c}}_{e, \hat{s}} \right] \\ & \geq (1 - \frac{4}{k \cdot m}) \cdot \sum_{\hat{c} \in \mathcal{C}} \sum_{\hat{s}=1}^k \left[ \sum_{(\hat{u}, \hat{s}) \in S_{\text{cur}}} p_{\hat{u}, \hat{s}}^{\hat{c}} x^{*\hat{c}}_{\hat{u}, \hat{s}} + \sum_{\substack{(\hat{u}, \hat{s)}, \\ (\hat{v}, \hat{s}) \in S_{\text{cur}}}} w_e^{\hat{c}} y^{*\hat{c}}_{e, \hat{s}} \right] \\ & = (1 - \frac{4}{k \cdot m}) \text{OPT}_{\text{LP}}(S_{\text{cur}}). \end{aligned}$$

Combining the above, we have

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

The lemma follows.  $\square$

Back to AVG-D, let the total number of iterations be  $T$ . Let  $S_{\text{tar}}^t$  and  $S_{\text{fut}}^t$  denote  $S_{\text{tar}}(c, s, x_{u,s}^{*c})$  and  $S_{\text{fut}}(c, s, x_{u,s}^{*c})$  in the  $t$ -th iteration of AVG-D, and  $S_{\text{cur}}^t = S_{\text{tar}}^t \cup S_{\text{fut}}^t$ . By Lemma 8,  $f(c, s, x_{u,s}^{*c}) = \max_{\hat{c}, \hat{s}, \hat{u}} f(\hat{c}, \hat{s}, x_{\hat{u}, \hat{s}}^{*\hat{c}}) \geq \mathbb{E}[f(c, s, \alpha)] \geq \frac{\text{OPT}_{\text{LP}}(S_{\text{cur}})}{4}$  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,

$$\begin{aligned} \text{ALG}(S_{\text{tar}}^1) + \frac{1}{4} \cdot \text{OPT}_{\text{LP}}(S_{\text{fut}}^1) &\geq \frac{1}{4} \cdot \text{OPT}_{\text{LP}}(S_{\text{cur}}^1) \\ \text{ALG}(S_{\text{tar}}^2) + \frac{1}{4} \cdot \text{OPT}_{\text{LP}}(S_{\text{fut}}^2) &\geq \frac{1}{4} \cdot \text{OPT}_{\text{LP}}(S_{\text{cur}}^2) = \frac{1}{4} \cdot \text{OPT}_{\text{LP}}(S_{\text{fut}}^1) \\ &\vdots \\ \text{ALG}(S_{\text{tar}}^T) + \frac{1}{4} \cdot \text{OPT}_{\text{LP}}(S_{\text{fut}}^T) &\geq \frac{1}{4} \cdot \text{OPT}_{\text{LP}}(S_{\text{fut}}^{T-1}). \end{aligned}$$

By summing up all the above inequalities, we have

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

which proves the approximation ratio.

Similar to that in AVG, a slight modification on AVG-D is to select only  $(c, s, x_{u,s}^{*c})$  that leads to a nonempty target group, so that  $T = O(nk)$ . We first prove that there does exist such  $(c, s, x_{u,s}^{*c})$  that  $f(c, s, x_{u,s}^{*c}) \geq \frac{1}{4} \cdot \text{OPT}_{\text{LP}}(S_{\text{cur}}(c, s, x_{u,s}^{*c}))$ . Consider those cases where  $\alpha =$

$x_{u,s}^{*c}$  is too large that the target group is empty. In this case  $S_{\text{cur}}(c, s, x_{u,s}^{*c}) = S_{\text{fut}}(c, s, x_{u,s}^{*c})$ , which implies  $f(c, s, x_{u,s}^{*c}) = \frac{1}{4} \cdot \text{OPT}_{\text{LP}}(S_{\text{cur}}(c, s, x_{u,s}^{*c}))$ , i.e., exactly the expected value. Suppose that no other  $(c, s, x_{u,s}^{*c})$  exists so that  $f(c, s, x_{u,s}^{*c}) \geq \frac{1}{4} \cdot \text{OPT}_{\text{LP}}(S_{\text{cur}}(c, s, x_{u,s}^{*c}))$ , then the expected value should be lower than  $\frac{1}{4} \cdot \text{OPT}_{\text{LP}}(S_{\text{cur}}(c, s, x_{u,s}^{*c}))$ , a contradiction. Thus, such  $(c, s, x_{u,s}^{*c})$  must exist, and AVG-D can always choose such  $(c, s, x_{u,s}^{*c})$ .

The number of iterations is thus also  $O(nk)$ . For each iteration, there are  $O(nmk)$  combinations of pivot parameters. Finding  $S_{\text{tar}}(c, s, x_{u,s}^{*c})$  and  $S_{\text{fut}}$ , 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  $(c, s, x_{u,s}^{*c})$ , finding  $f(c, s, x_{u,s}^{*c})$  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  $O(\text{LP}) + O(n^2 m k^2 |E|)$ . 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  $f(c, s, x_{u,s}^{*c})$  can achieve at most a speedup factor of  $nmk$ , which further lowers the time complexity to  $O(\text{LP}) + O(nmk|E|)$ .  $\square$

## 4.4 Enhancements

In the following, we detail enhancements of the AVG and AVG-D algorithms. First, we show that AVG and AVG-D support values of  $\lambda \neq \frac{1}{2}$  via a simple scaling on the inputs. We then design two advanced strategies, including an *advanced LP transformation technique* and a *new focal parameter sampling scheme*, to improve the efficiency of AVG and AVG-D. The LP transformation technique derives a new LP formulation to reduce the number of decision variables and constraints from  $O((n + |E|)mk)$  to  $O((n + |E|)m)$  by condensing the  $x_{u,s}^c$  variables ( $k$  is the number of slots). We prove that the optimal objective in the new formulation is exactly that in the original one. The focal parameter sampling scheme maintains a *maximum utility factor*  $x_s^{*c}$  (detailed later) for each pair of item  $c$  and slot  $s$  to avoid unnecessary sampling of focal parameters  $(c, s, \alpha)$  when  $\alpha \geq \max_{u \in V} x_{u,s}^c$ , especially for a large  $k$ . We prove that the sampling results of the new sampling technique and the original one are the same. Therefore, the efficiency of AVG can be improved without sacrificing the solution quality. Finally, we extend AVG and AVG-D to support SVGIC-ST (and also the Social Event Organization (SEO)-type problems) by tailoring CSF with consideration of the additional VR-related constraints.

**Supporting Other Values of  $\lambda$ .** First, observe that  $\lambda = 0$  corresponds to a special case of maximizing only the total preference utility, where a simple greedy algorithm can find the exact optimal solution. Assume that  $\lambda \neq 0$ . To support the cases with  $\lambda \neq \frac{1}{2}$ , AVG first sets the *scaled preference values* for each  $(u, c)$  as  $p'(u, c) = \frac{1-\lambda}{\lambda} p(u, c)$ . This value is then used instead of  $p(u, c)$  in the LP relaxation problem, as well as in the computation of  $f(c, s, x_{u,s}^{*c})$  in

DPS in AVG-D. Since

$$\begin{aligned} & (1 - \lambda) \cdot p(u, c) + \lambda \cdot \sum_{v|u \xleftrightarrow{c} v} \tau(u, v, c) \\ &= 2\lambda \cdot \left( \frac{1}{2} \cdot p'(u, c) + \frac{1}{2} \cdot \sum_{v|u \xleftrightarrow{c} v} \tau(u, v, c) \right), \end{aligned}$$

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

**Advanced LP Transformation.** Recall that the LP relaxation of the SVGIC problem (denote by  $\text{LP}_{\text{SVGIC}}$ ) is

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

subject to the following constraints:

$$\begin{aligned} & \sum_{s=1}^k x_{u,s}^c \leq 1, \quad \forall u \in V, c \in \mathcal{C} \\ & \sum_{c \in \mathcal{C}} x_{u,s}^c = 1, \quad \forall u \in V, s \in [k] \\ & x_u^c = \sum_{s=1}^k x_{u,s}^c, \quad \forall u \in V, c \in \mathcal{C} \\ & y_e^c = \sum_{s=1}^k y_{e,s}^c, \quad \forall e = (u, v) \in E, c \in \mathcal{C} \\ & y_{e,s}^c \leq x_{u,s}^c, \quad \forall e = (u, v) \in E, s \in [k], c \in \mathcal{C} \\ & y_{e,s}^c \leq x_{v,s}^c, \quad \forall e = (u, v) \in E, s \in [k], c \in \mathcal{C} \\ & x_{u,s}^c, x_u^c, y_{e,s}^c, y_e^c \geq 0, \quad \forall u \in V, e \in E, s \in [k], c \in \mathcal{C}. \end{aligned}$$

Solving  $\text{LP}_{\text{SVGIC}}$ , which is the first step of the AVG algorithm, deals with  $O((n + |E|)mk)$  decision variables and  $O((n + |E|)mk)$  constraints. In the following, we transform  $\text{LP}_{\text{SVGIC}}$  into a more compact linear program formulation  $\text{LP}_{\text{SIMP}}$  with fewer decision variables as follows.

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

subject to the following constraints:

$$\begin{aligned} & \mathbf{x}_u^c \leq 1, \quad \forall u \in V, c \in \mathcal{C} \\ & \sum_{c \in \mathcal{C}} \mathbf{x}_u^c = k, \quad \forall u \in V \\ & \mathbf{y}_e^c \leq \mathbf{x}_u^c, \quad \forall e = (u, v) \in E, c \in \mathcal{C} \\ & \mathbf{y}_e^c \leq \mathbf{x}_v^c, \quad \forall e = (u, v) \in E, c \in \mathcal{C} \\ & \mathbf{x}_u^c, \mathbf{y}_e^c \geq 0, \quad \forall u \in V, e \in E, c \in \mathcal{C}. \end{aligned}$$

To avoid confusion, we use normal math symbols ( $x$  and  $y$ ) to represent decision variables in  $\text{LP}_{\text{SVGIC}}$  and Typewriter symbols ( $\mathbf{x}$  and  $\mathbf{y}$ ) to denote those in  $\text{LP}_{\text{SIMP}}$ . Let  $\text{OPT}_{\text{SIMP}}$  and  $\text{OPT}_{\text{SVGIC}}$  denote the optimal objective values in  $\text{LP}_{\text{SIMP}}$  and  $\text{LP}_{\text{SVGIC}}$ . We have the following observation.

**Observation 2.**  $\text{OPT}_{\text{SIMP}} = \text{OPT}_{\text{SVGIC}}$  always holds. Moreover, given an optimal solution  $\mathbf{X}^* = \{\mathbf{x}_u^c\}$  for  $\text{LP}_{\text{SIMP}}$ , there exists an optimal solution  $X^*$  of  $\text{LP}_{\text{SVGIC}}$  with  $x_{u,s}^{*c} = \frac{1}{k} \cdot \mathbf{x}_u^c, \forall c, u, s$ .

*Proof.* We first prove  $\text{OPT}_{\text{SIMP}} \leq \text{OPT}_{\text{SVGIC}}$ . Given any feasible solution  $\mathbf{X}$  of  $\text{LP}_{\text{SIMP}}$ , let  $X$  denote a solution of  $\text{LP}_{\text{SVGIC}}$  such that  $x_{u,s}^c = \frac{1}{k} \cdot \mathbf{x}_u^c, \forall c, u, s$ . By construction, we have  $x_u^c = \sum_{s=1}^k x_{u,s}^c = k \cdot \frac{1}{k} \cdot \mathbf{x}_u^c = \mathbf{x}_u^c$  for all  $c, u, s$ , and analogously  $y_e^c = \mathbf{y}_e^c$  for all  $c, e$ . Therefore,  $X$  is feasible in  $\text{LP}_{\text{SVGIC}}$ . Moreover, it achieves the same objective value in  $\text{LP}_{\text{SVGIC}}$  as  $\mathbf{X}$  in  $\text{LP}_{\text{SIMP}}$ , which implies  $\text{OPT}_{\text{SIMP}} \leq \text{OPT}_{\text{SVGIC}}$ . We then prove  $\text{OPT}_{\text{SIMP}} \geq \text{OPT}_{\text{SVGIC}}$ . Given any feasible solution  $X$  of  $\text{LP}_{\text{SVGIC}}$ , let  $\mathbf{X}$  denote a solution of  $\text{LP}_{\text{SIMP}}$  such that  $\mathbf{x}_u^c = x_u^c = \sum_{s=1}^k x_{u,s}^c$ . Similarly,  $\mathbf{X}$  is a feasible solution of  $\text{LP}_{\text{SIMP}}$ , and it achieves exactly the same objective value in  $\text{LP}_{\text{SIMP}}$  as  $X$  in  $\text{LP}_{\text{SVGIC}}$ . Therefore,  $\text{OPT}_{\text{SIMP}} \geq \text{OPT}_{\text{SVGIC}}$ . Combining the above, we have  $\text{OPT}_{\text{SIMP}} = \text{OPT}_{\text{SVGIC}}$ , and the second part of the observation follows from the above construction.  $\square$

As AVG requires only *one* optimal solution for  $\text{LP}_{\text{SVGIC}}$ , it suffices for AVG to first solve the above simplified linear program  $\text{LP}_{\text{SIMP}}$  with only  $O((n + |E|)m)$  decision variables and constraints. AVG then scales the optimal solution by a factor of  $\frac{1}{k}$  according to Observation 2 to find an optimal solution for  $\text{LP}_{\text{SVGIC}}$ . Therefore,  $\text{LP}_{\text{SIMP}}$  effectively improves the efficiency of AVG.

**Advanced Focal Parameter Sampling.** In AVG, a set of focal parameters  $(c, s, \alpha)$  is sampled uniformly at random for each iteration. However, for input instances with a large  $k$ , the utility factor  $x_{u,s}^{*c}$  of assigning item  $c$  to user  $u$  at slot  $s$  tends to become smaller, e.g., roughly  $O(\frac{1}{k})$ , especially after the above LP transformation strategy is applied. Recall that AVG compares  $\alpha$  to every user eligible for viewing item  $c$  at slot  $s$  and displays  $c$  to user  $u$  if  $x_{u,s}^{*c} \geq \alpha$ . If  $\alpha > \max_{u \in V} x_{u,s}^{*c}$ , AVG essentially remains idle (i.e., making no progress) in the current round and needs to re-sample another set of focal parameters in the next iteration. For instances with large  $k$ , it is thus more difficult for AVG to effectively sample a set of *good* focal parameters without the above property.

To address the above issue, we design an advanced focal parameter sampling scheme as follows. For each pair of  $(c, s)$ , AVG maintains the *maximum utility factor*  $\bar{x}_s^{*c} = \max_{u \in V} \{x_{u,s}^{*c} | u \text{ is eligible for } (c, s)\}$ . Specifically,  $\bar{x}_s^{*c}$  is initialized as  $\max_{u \in V} x_{u,s}^{*c}$  after  $X^*$  (the optimal solution of LP) is retrieved, and it is constantly updated along the rounding process (as the number of users eligible for  $(c, s)$  decreases). In each iteration, AVG first randomly samples a pair  $(c, s)$  with probability  $\frac{\bar{x}_s^{*c}}{\sum_{c \in \mathcal{C}, s \in [k]} \bar{x}_s^{*c}}$ , i.e., proportional to the value  $\bar{x}_s^{*c}$ . It then samples  $\alpha$  uniformly at random from  $[0, \bar{x}_s^{*c}]$ .

Specifically, let **GOOD** denote the event that a set of good focal parameters is sampled, i.e.,  $\alpha$  is at most  $\bar{x}_s^{*c}$  for the selected  $(c, s)$  in the original sampling scheme. Moreover, for a set of good focal parameter  $(c, s, \alpha)$ , let **TARGET** $(c, s, \alpha)$  denote the event that Co-display Subgroup Formation ends up with the same behavior as if  $(c, s, \alpha)$  is sampled. We have the following observation. Let  $\text{Pr}_{\text{orig}}(\text{TARGET}(c, s, \alpha))$  denote the probability that **TARGET** $(c, s, \alpha)$  happens in the original sample scheme, and let  $\text{Pr}_{\text{adv}}(\text{TARGET}(c, s, \alpha))$  denote the probability of **TARGET** $(c, s, \alpha)$  in the advanced sample scheme. The following observation indicates that sampling with the advanced sampling scheme is equivalent (in terms of outcome distribution) to sampling *good* parameters, i.e.,

picking  $(c, s, \alpha)$  directly and only from the pool of focal parameters that lead to a successful assignment of at least one item to one user at some slot.

**Observation 3.** For any set of good focal parameter  $(c, s, \alpha)$  such that  $\alpha \leq \max_{u \in V} x_{u,s}^*$ ,  $\Pr_{\text{adv}}(\text{TARGET}(c, s, \alpha)) = \Pr_{\text{orig}}(\text{TARGET}(c, s, \alpha)|\text{GOOD})$ .

*Proof.* By construction,  $\Pr_{\text{orig}}(\text{GOOD}) = \sum_{c \in \mathcal{C}, s \in [k]} \frac{1}{k \cdot m} (\bar{x}_s^*)$ .

Next, by construction, it is straightforward that the advanced sampling scheme only samples good focal parameters. Consider a set of good focal parameter  $(c, s, \alpha)$ . Let  $u_1$  and  $u_2$  denote the users such that  $u_1 = \arg \max_{x_{u,s}^* < \alpha} x_{u,s}^*$

and  $u_2 = \arg \min_{x_{u,s}^* \geq \alpha} x_{u,s}^*$ . By definition, if CSF chooses any  $\alpha \in (x_{u_1,s}^*, x_{u_2,s}^*]$ , it ends up with the same behavior as if  $(c, s, \alpha)$  is sampled, i.e., TARGET happens. With  $(c, s)$  given, this happens with probability  $p_\alpha = x_{u_2,s}^* - x_{u_1,s}^*$ . Note that  $u_2$  is guaranteed to exist, and we replace  $x_{u_1,s}^*$  with 0 if  $u_1$  does not exist (i.e., the case where every eligible user is assigned  $c$  by Co-display Subgroup Formation). In the advanced sampling scheme, the probability of this event is thus  $\Pr_{\text{adv}}(\text{TARGET}(c, s, \alpha)) = \frac{\bar{x}_s^*}{\sum_{c \in \mathcal{C}, s \in [k]} \bar{x}_s^*} \cdot p_\alpha$ . On the

other hand, in the original sampling scheme, the probability of this event is  $\Pr_{\text{orig}}(\text{TARGET}(c, s, \alpha)) = \frac{1}{k \cdot m} \cdot p_\alpha$ . We have

$$\begin{aligned} \Pr_{\text{orig}}(\text{TARGET}(c, s, \alpha)|\text{GOOD}) &= \frac{\Pr_{\text{orig}}(\text{TARGET}(c, s, \alpha))}{\Pr_{\text{orig}}(\text{GOOD})} \\ &= \frac{\frac{1}{k \cdot m} \cdot p_\alpha}{\sum_{c \in \mathcal{C}, s \in [k]} \frac{1}{k \cdot m} (\bar{x}_s^*)} \\ &= \frac{p_\alpha}{\sum_{c \in \mathcal{C}, s \in [k]} \bar{x}_s^*} \\ &= \Pr_{\text{adv}}(\text{TARGET}(c, s, \alpha)). \end{aligned}$$

The observation follows.  $\square$

Therefore, the advanced focal sampling scheme effectively improves the efficiency of AVG since it discards the non-contributing sampling events in advance. Note that it can also be applied to improve AVG-D such that the events of selecting non-controlling focal parameters will not be simulated in the derandomization process. The main algorithm AVG with the advanced LP transformation strategy and focal sampling scheme are detailed in Algorithm 4. The effects of these two speedup techniques are tested in Section 6.4.

**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 3.3, 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  $x_{u,s}^* \geq \alpha$ , 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  $x_{u,s}^* = 0$  for all remaining eligible users, as well as removes  $(c, s, \alpha)$  from the set of candidate parameter

---

**Algorithm 4** AVG with advanced LP transformation and focal parameter sampling

---

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

- 1:  $\mathbf{A}(\hat{u}, \hat{s}) \leftarrow \text{NULL}$  for all  $\hat{u}, \hat{s}$
- 2: Construct  $\text{LP}_{\text{SVGIC}}$  (Original Relaxed LP)
- 3: Transform into  $\text{LP}_{\text{SIMP}}$  (Simplified LP)
- 4:  $\bar{X} \leftarrow$  optimal solution for  $\text{LP}_{\text{SIMP}}$
- 5:  $\bar{x}_s^* \leftarrow \max_{u \in V} x_{u,s}^*$
- 6: **for**  $u \in V, c \in \mathcal{C}, s \in [k]$  **do**
- 7:      $x_{c,s}^* \leftarrow \frac{1}{k} \cdot \bar{x}_u^*$
- 8: **while** some entry in  $\mathbf{A}$  is NULL **do**
- 9:     Sample  $(c, s)$  randomly with probability  $\frac{\bar{x}_s^*}{\sum_{c \in \mathcal{C}, s \in [k]} \bar{x}_s^*}$
- 10:    Sample  $\alpha \in [0, \bar{x}_s^*]$  uniformly at random
- 11:    **for**  $\hat{u} \in V$  **do**
- 12:      **if**  $\mathbf{A}(\hat{u}, s) = \text{NULL}$  and  $\mathbf{A}(\hat{u}, t) \neq c \forall t \neq s$  **then**  
                ( $\hat{u}$  eligible for  $(c, s)$ )
- 13:        **if**  $x_{\hat{u},s}^* \geq \alpha$  **then**
- 14:           $\mathbf{A}(\hat{u}, s) \leftarrow c$
- 15:        Update  $\bar{x}_s^*$
- 16:    **return**  $\mathbf{A}$

---

sets in AVG-D. Therefore, AVG never displays more than  $M$  users to the same item in the final SAVG  $k$ -Configuration.

**Supporting Social Event Organization.** We have also identified *Social Event Organization* (SEO) as another important application of the targeted problem. Specifically, SEO [14, 46, 69, 70, 85] is a line of research that studies the problem of organizing overlapping and conflicting events for users of Event-Based Social Network (EBSN) (e.g. Meetup [56], Facebook Events [21], and Douban Location [17]). Given a set of users and a set of events, SEO assigns each user to a single or a series of events, such that the total preferences of the selected users to their assigned events are maximized, while several constraints are satisfied, e.g. traveling cost budget, time conflict avoidance, and event sizes. Consequently, the optimization problems for SEO are closely related to SVGIC: social event attendees directly correspond to VR shopping users, while social events correspond to the displayed items in VR. SVGIC can be viewed as a general version of the SEO problem that incorporates both social-based utilities to support a series of social events. It is then straightforward to see the above extension for SVGIC-ST supports SEO, as the social event size constraints can be modelled as subgroup size constraints.

## 5. EXTENSIONS FOR PRACTICAL SCENARIOS

In this section, we extend SVGIC and AVG to support a series of practical scenarios. 1) *Commodity values*. Each item is associated with a commodity value to maximize the total profit. 2) *Layout slot significance*. Each slot location is associated with a different significance weight (e.g., center is better) according to retailing research [18, 77]. 3) *Multi-View Display*, where a user can be displayed multiple items in a slot, including one default, personally preferred item in the primary view and multiple items to view with friends in group views, and the primary and group views can be freely switched. 4) *Generalized social benefits*, where social utility

can be measured not only pairwise (each pair of friends) but also group-wise (any group of friends). 5) *Subgroup change*, where the fluctuations (i.e., change of members) between the partitioned subgroups at consecutive slots are limited to ensure smooth social interactions as the elapse of time. 6) *Dynamic scenario*, where users dynamically join and leave the system with different moving speeds.

**A. Commodity Value.** The current objective function, i.e., maximizing the total utility among all users, is not designed from a retailer's perspective. In contrast, it will be more profitable to maximize the *total expected profit* by taking into account the commodity values, while regarding the final SAVG utility as the willingness/likelihood of purchases. Therefore, the ILP formulation of SVGIC and AVG can be modified to examine the total SAVG utility weighted by a commodity value  $\omega_c$  for each item  $c$ . Let  $\omega_c$  denote the commodity price of each item  $c$ . The new objective of SVGIC is the weighted total SAVG utility as follows.

$$\sum_{u \in V} \sum_{s=1}^k \omega_{\mathbf{A}^*(u,s)} \cdot [p(u, \mathbf{A}^*(u,s)) + \sum_{v|u \xleftarrow{\mathbf{A}^*(u,s)} v} \tau(u, v, \mathbf{A}^*(u,s))].$$

**B. Layout Slot Significance.** Retailing research [18, 77] manifests that different slot locations in the item placement layout have varying significance for users. For example, the slots at the center of an aisle are nine times more important than the slots at the ends [77]. Vertical slots right at the eye-level of users are also more significant [18]. Therefore, it is important for SVGIC to not only select the items but also optimize the corresponding placements on the shelf. Specifically, let  $\gamma_s$  represent the relative significance of slot  $s$  in the layout. The new objective of SVGIC is the SAVG utility, weighted by the relative significance as follows.

$$\sum_{u \in V} \gamma_s \cdot \sum_{s=1}^k [p(u, \mathbf{A}^*(u,s)) + \sum_{v|u \xleftarrow{\mathbf{A}^*(u,s)} v} \tau(u, v, \mathbf{A}^*(u,s))].$$

**C. Multi-View Display (MVD).** SAVG  $k$ -Configuration with CID allows each user to see only *one* item at each slot. In contrast, MVD [33, 48] has been proposed to support the display of multiple items at the same slot, so that the primary view (for a personally preferred item) and the group view (for common items with friends) can be freely switched. Let  $\beta$  denote the maximum number of items (i.e., views) displayed for each user in each slot (depending on the user interface). an *MVD-supportive* SAVG  $k$ -Configuration  $\mathbf{A}$  maps a tuple  $(u, s)$  to an itemset  $\mathbf{A}(u, s)$  with at most  $\beta$  items, and the new objective function of SVGIC is as follows.

$$\sum_{u \in V} \sum_{s=1}^k (\sum_{c \in \mathbf{A}^*(u,s)} [p(u, c) + \sum_{v|u \xleftarrow{c} v} \tau(u, v, c)])$$

**D. Generalized Social Benefits.** In SVGIC  $\tau(u, v, c)$  considers only the *pair-wise* social interaction, i.e., social discussion and influence between two users viewing the common item. However, existing research on social influence [4, 36, 100] manifests that quantifying the magnitude of social influence from *group-wise* interactions is more general since

the pair-wise interaction is a special case of the group-wise interaction. Therefore, let  $\tau(u, \mathcal{V}, c)$  denote the subgroup social utility for user  $u$  when co-displaying  $c$  to a *subgroup of friends*  $\mathcal{V}$  learned from group-wise social interaction models such as [36]. Let  $u \xleftrightarrow{s} \mathcal{V}$  represent the *maximal group co-display*<sup>3</sup>, i.e., user  $u$  sees item  $c$  at slot  $s$  with every user in  $\mathcal{V}$ , and no friend of  $u$  outside  $\mathcal{V}$  also sees  $c$  at slot  $s$ . Thus, the new objective of SVGIC is generalized as follows.

$$\sum_{u \in V} \sum_{s=1}^k [p(u, \mathbf{A}^*(u,s)) + \sum_{\mathcal{V}|u \xleftrightarrow{s} \mathcal{V}} \tau(u, \mathcal{V}, \mathbf{A}^*(u,s))].$$

**Algorithm Extension for Scenarios A–D.** In the following, we first introduce a new ILP formulation to support the above practical scenarios. Decision variable  $x_{u,s}^c$  now indicates whether user  $u$  views item  $c$  in the *primary view* at slot  $s$ . On the other hand, let decision variable  $w_{u,s}^c$  denote whether  $u$  can see  $c$  in *some view*, i.e., in the primary view or a group view, at slot  $s$ . Therefore,  $x_{u,s}^c = 1$  implies  $w_{u,s}^c = 1$ . To consider the layout slot significance, let  $w_u^c = \sum_{s=1}^k \gamma_s \cdot w_{u,s}^c$  be the weighted sum of all  $w$  decision variables, and here  $w_u^c$  is not required to be binary. Therefore,  $w_u^c$  represents the total significance of the assigned slots for  $u$  to view  $c$ . To capture the maximal co-display, for all subgroups of users  $\mathcal{V} \subseteq V$ , let  $y_{u,\mathcal{V},s}^c$  now indicate the *maximal group co-display*. In other words,  $u \xleftrightarrow{s} \mathcal{V}$  if and only if  $y_{u,\mathcal{V},s}^c = 1$ . Analogously, let  $y_{u,\mathcal{V}}^c$  be the weighted sum of  $y_{u,\mathcal{V},s}^c$ . The new objective function is as follows.

$$\max \quad \sum_{c \in \mathcal{C}} \omega_c \cdot [\sum_{u \in V} (p(u, c) \cdot w_u^c + \sum_{\mathcal{V} \subseteq V} (\tau(u, \mathcal{V}, c) \cdot y_{u,\mathcal{V}}^c))]$$

subject to the constraints,

$$\sum_{c \in \mathcal{C}} x_{u,s}^c = 1, \quad \forall u \in V, s \in [k] \quad (11)$$

$$\sum_{c \in \mathcal{C}} w_{u,s}^c \leq \beta, \quad \forall u \in V, s \in [k] \quad (12)$$

$$x_{u,s}^c \leq w_{u,s}^c, \quad \forall u \in V, s \in [k], c \in \mathcal{C} \quad (13)$$

$$\sum_{s=1}^k x_{u,s}^c \leq 1, \quad \forall u \in V, c \in \mathcal{C} \quad (14)$$

$$x_u^c = \sum_{s=1}^k (\gamma_s \cdot x_{u,s}^c), \quad \forall u \in V, c \in \mathcal{C} \quad (15)$$

$$y_{\mathcal{V}}^c = \sum_{s=1}^k (\gamma_p \cdot y_{\mathcal{V},s}^c), \quad \forall \mathcal{V} \subseteq V, c \in \mathcal{C} \quad (16)$$

$$y_{\mathcal{V},s}^c \leq x_{u,s}^c, \quad \forall u \in \mathcal{V}, s \in [k], c \in \mathcal{C} \quad (17)$$

$$y_{\mathcal{V},s}^c + x_{u,s}^c \leq 1, \quad \forall u \notin \mathcal{V}, s \in [k], c \in \mathcal{C} \quad (18)$$

$$x_{u,s}^c, y_{\mathcal{V},s}^c \in \{0, 1\}, \quad \forall u \in V, e \in E, s \in [k], c \in \mathcal{C}. \quad (19)$$

Constraint (11) guarantees that a user is displayed exactly one default item (for the primary view) at each slot.

<sup>3</sup>Here only the maximal groups, instead of every subgroup, are examined in the objective function; otherwise, duplicated social utilities will be summed up because multiple overlapping subgroups are contained in a maximal groups

Constraint (12) ensures that at most  $\beta$  items can be viewed at one slot in MVD. Constraint (13) states that the default item can be viewed in the MVD, i.e., if  $c$  is displayed ( $x_{u,s}^c = 1$ ), then  $w_{u,s}^c$  will become 1. Constraint (14) guarantees that the default displayed items for each user are not replicated. Constraint (15) ensures that  $x_u^c$  (significance of the slot where  $u$  is displayed  $c$ ) is the weighted sum of  $x_{u,s}^c$ . Similarly, constraint (16) states that  $y_{\mathcal{V},s}^c$  is the weighted sum of  $y_{\mathcal{V},s}^c$  (on the slot significance). Constraints (17) and (18) specify the maximal co-display. Constraint (17) states that  $y_{\mathcal{V},s}^c = 1$  only if every  $x_{u,s}^c = 1$ , implying that  $c$  is displayed to every  $u$  in  $\mathcal{V}$  at slot  $s$ . When  $y_{\mathcal{V},s}^c = 1$ , constraint (18) ensures no other  $x_{u,s}^c$  is 1 for  $u$  outside of  $\mathcal{V}$ , so that  $\mathcal{V}$  is indeed the maximal set. Finally, constraint (19) ensures the decision variables to be binary, i.e.,  $\in \{0, 1\}$ .

The LP relaxation of the above formulation also relaxes the last constraint, so that binary decision variables  $x_{u,s}^c, y_{\mathcal{V},s}^c$  become real numbers in  $[0, 1]$ . The other two phases of AVG and AVG-D remain unchanged. The commodity value  $\omega_c$  and the layout slot significance  $\gamma_s$  will not affect the theoretical guarantee of AVG and AVG-D because the original proof can be modified by replacing the preference utility and social utility values with weighted versions. Furthermore, in any iteration  $t$  in AVG, for any subgroup of users  $\mathcal{V}$  with all users  $u \in \mathcal{V}$  eligible for  $(c, s)$ , as the maximal co-displayed subgroups contain at most  $(\max |\mathcal{V}|)$  users, Lemma 7 can be analogously modified to upper bound the probability that *at least one* user in  $\mathcal{V}$  is not eligible for  $(c, s)$  in iteration  $(t + 1)$  by summing exactly  $2(\max |\mathcal{V}|)$  summation terms as in the last inequality in the proof of Lemma 7. Consequently, AVG and AVG-D can achieve a  $2(\max |\mathcal{V}|)$ -approximation. Moreover, the advanced focal parameter sampling scheme can be employed in the above model.

**E. Subgroup Changes.** While the partitions of subgroups in AVG enable flexible configurations and facilitates social discussion, drastic subgroup changes between adjacent slots may undermine the smoothness of social interactions and discussions within the group when users are exploring the VE. To address this issue, one approach is to incorporate a constraint on *edit distances* between the partitioned induced subgroups at consecutive slots. Specifically, a pair of friends  $u$  and  $v$  in the same subgroup (i.e., viewing a common item) at slot  $s$  but separated into different subgroups at slot  $s + 1$  will contribute 1 to the edit distance between the two slots. To reduce the subgroup changes, IP of the extended AVG discourages the intermediate solution (the fractional solution) to introduce large subgroup changes by constraining the total edit distances across all consecutive slots in the SAVG  $k$ -Configuration. In the rounding scheme, AVG examines the potential outcomes before co-displaying a focal item to the target subgroup. It voids the current iteration and re-sample another set of pivot parameters, if the solution incurs a large subgroup change. Finally, after the complete SAVG  $k$ -Configuration is constructed, AVG can facilitate a local search by exchanging the sub-configuration at different slots to reduce the total subgroup change.

**F. Dynamic Scenario.** To support the dynamic join and leave of users with different walking speeds, executing the whole AVG for every dynamic event is computationally intensive. Therefore, one approach is to first solve the relaxed LP for the partial SVGIC instance with the new setting, whereas the majority of utility factors remain the

same (temporarily regarded as constants in LP), so that the utility factors are updated efficiently. CSF then assigns new users to existing target subgroups according to their utility factors, and it also invokes local search to examine the potential profit from exchanging users among existing target subgroups. Moreover, similar to in SVGIC-ST, *teleportation* in VR enables a user to move directly to her friends at a different slot. Therefore, when two users are co-displayed an item (directly or indirectly) but are far apart from each other in the store, the shopping application could suggest one of them to teleport to the other's location.

## 6. 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.

### 6.1 Experiment Setup and Evaluation Plan

**Datasets.** To evaluate the proposed algorithms, three real datasets are tested in the experiment. The first dataset *Timik* [40] 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* [20], is a website containing the reviews of 139K products and a social trust network with 132K users and 841K edges. The third dataset, *Yelp* [90], 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 [7, 27, 71] 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 [47] which jointly models the social influence between users and the latent topics of items. Following the scales of the experiments in previous research [65, 67], the default number of slots  $k$ , number of items  $m$ , and size of user set  $n$  selected from the social networks to visit a VR store are set as 50, 10000, and 125, respectively.

**Baseline Methods.** We compare AVG and AVG-D with five baseline algorithms: Personalized Top- $k$  (PER), Fairness Maximization in Group recommendation (FMG) [67], Social-aware Diverse and Preference selection (SDP) [71], Group Recommendation and Formation (GRF) [65], 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 3.3 that finds the optimal solutions of small SVGIC instances by Gurobi [29]. 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.

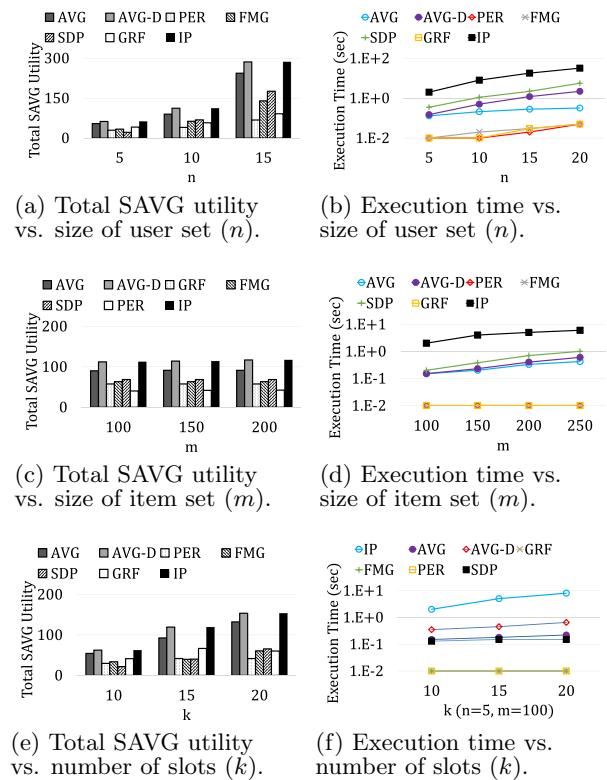
**Evaluation Metrics.** To evaluate the algorithms considered for SVGIC and analyze the returned SAVG  $k$ -Configurations, we introduce the following metrics: 1) total SAVG utility achieved, 2) total execution time (in seconds), 3) the percentages of personal preference utility (*Personal%*) and social 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-display%*), 7) the percentage of users viewing items alone (*Alone%*), and 8) *regret ratio* (a fairness measure detailed later in Section 6.5.) 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.

**Evaluation Plan.** To evaluate the performance of the above algorithms, in Section 6.2, we use IP to derive the optimal total SAVG utility on small datasets.<sup>4</sup> 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 [57]. Next, in Section 6.3, we evaluate the efficacy of AVG and AVG-D in large datasets with input scales following previous research [65, 67], while conducting experiments on different inputs generated by PIERT [47] (default), AGREE, and GREE [9]. We examine the efficiency of all algorithms, including various configurations of mixed integer programming (MIP) algorithms, in Section 6.4. We then compare the algorithms on the aforementioned group formation-related performance metrics in Section 6.5. A case study on a 2-hop ego network in Yelp is shown in Section 6.6 to examine the subgroup partition patterns in different algorithms. Result of a sensitivity test on  $r$ , an important algorithmic parameter of the deterministic AVG-D algorithm and experimental results on the SVGIC-ST problem are reported in Sections 6.7 and 6.8, respectively. Finally, we build a prototype of VR store with Unity 2017.1.1 (64bit), Photon Unity Network free 1.86, SteamVR Plugin 1.2.2, VRTK, and 3ds Max 2016 for hTC VIVE HMD to validate the proposed objective in modelling SVGIC. We detail the user study in Section 6.9.

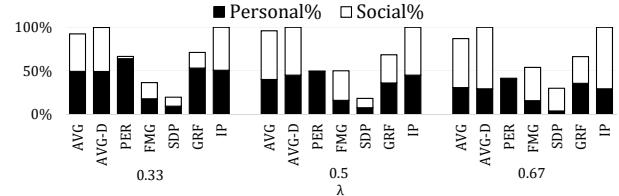
## 6.2 Comparisons on Small Datasets

Figure 3(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

<sup>4</sup>Since solving IP is NP-hard, Gurobi cannot solve large instances within hours.



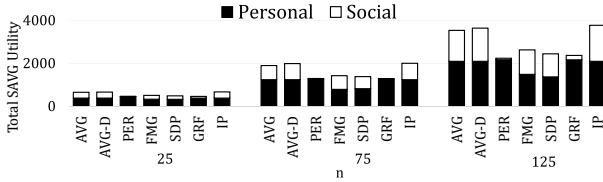
**Figure 3: Comparisons on small datasets.**



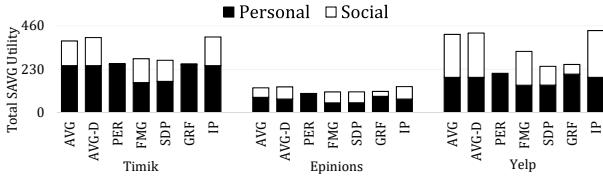
**Figure 4: Normalized total SAVG utility of diff.  $\lambda$ .**

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 3(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 3(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 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 3(b), 3(d) and 3(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



**Figure 5: Total SAVG utility vs. size of user set ( $n$ ).**



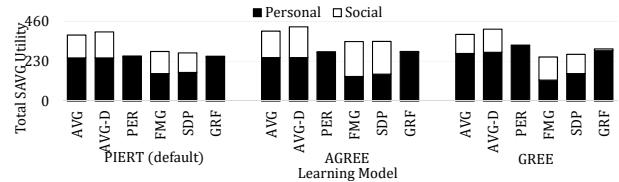
**Figure 6: Total SAVG utility in diff. datasets.**

baseline approaches focus only on one of preference utility, social utility, or subgroup partition, instead of considering all of them jointly. Figure 4 evaluates the impact of  $\lambda$  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  $\lambda$  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.

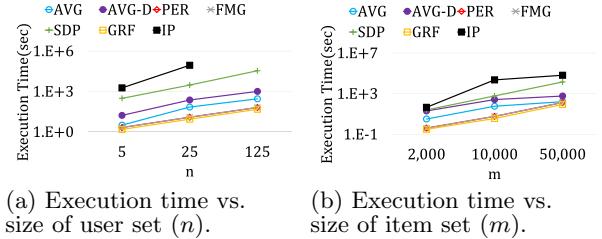
### 6.3 Sensitivity Tests on Large Datasets

We evaluate the efficiency and efficacy of AVG and AVG-D in large datasets with the scales of the three dimensions following previous research [65, 67], i.e.,  $m = 10000$ ,  $k = 50$ , and  $n = 125$ . Figure 5 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. Moreover, the returned objective values of AVG and AVG-D respectively achieve at least 93.7% and 96.4% of the objective value of IP, manifesting that our algorithms are effective. 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.

Figure 6 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 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



**Figure 7: Total SAVG utility vs. different input.**



**Figure 8: Execution time in Yelp.**

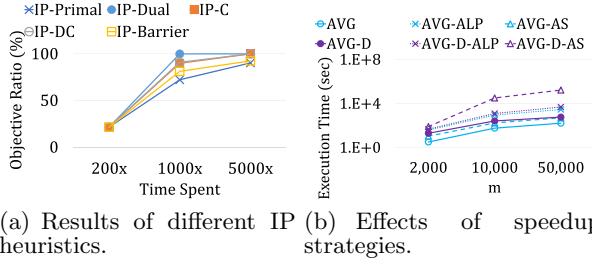
the social utility is lower.

Next, to examine the influence of input models on the tackled problem, Figure 7 shows the experimental results on different inputs generated by PIERT [47] (default), AGREE and GREE [9]. PIERT jointly learns the preference and social utilities by modeling the social influence between users and the latent topics of items. For AGREE and GREE [9], the former assumes the social influence between users is equal, and the later learns sophisticated weights of the triple (user, user, item). AVG and AVG-D outperform the baselines with regards to all different input models, manifesting that our method is generic to different distributions of inputs. Note that the social utilities returned by AVG and AVG-D with PIERT and AGREE are slightly greater than the ones with GREE. The result manifests that AVG and AVG-D can select better items for users to enjoy if social utilities are different across items.

### 6.4 Scalability Tests on Large Datasets

Figure 8(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 8(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 [83] and IrisVR [39], seldom have  $n > 25$ .

AVG, AVG-D, and IP are all integer program (IP)-based approaches, which means solving an IP (or a relaxed linear program) is the first step. For a fixed  $k$  (number of display slots for a user), when  $n$  and  $m$  grow, the numbers of variables and constraints in IP and LP also increase. However, as the number of *core decisions* (which items to be displayed in the  $k$  slots) remains the same, an increasing number of newly introduced constraints become *dominated* (i.e., redundant) and thus pre-solved (relaxed) by the highly optimized commercial IP/LP solvers. Accordingly, the em-



**Figure 9: Sensitivity Test on Efficiency and Efficacy.**

pirical complexity of the problem models *after* pre-solving will not grow explicitly with  $n$  and  $m$ , rendering the practical computational cost of solving the IP/LP more scalable regarding  $n$  and  $m$ . Moreover, in AVG and AVG-D, after the optimal fractional solution for the LP is retrieved, decision variables with values of 0 or 1 do not require the randomized/deterministic rounding procedure and can be directly filled into the final solution. This usually happens when 1) some user has a very strong preference toward a specific item such that viewing the item by her own still outweighs viewing other items with friends, and 2) some item is extremely popular and thus is co-displayed to a majority of users. As  $n$  and  $m$  increases in this case, the total number of decision variables  $x^*(u, \cdot)$  grows, but the summation of all decision variables remains constant in Constraint (2). Therefore, more decision variables have values close to 0 in this case. On the other hand, all other baselines are based on local greedy heuristics that need either a linear scan on all users ( $O(n)$ ) or all items ( $O(m)$ ) at each step, and do not benefit from the above “decision dilution” phenomenon.

To examine the performance of different mixed integer programming (MIP) algorithms, we further conducted experiments with Primal-first Mixed Integer Programming (IP-Primal), Dual-first Mixed Integer Programming (IP-Dual), Concurrent Mixed Integer Programming (IP-C), Deterministic Concurrent Mixed Integer Programming (IP-DC), and the Barrier Algorithm (IP-Barrier) in the Gurobi [29] package. Figure 9(a) shows the trade-off between efficiency and efficacy of five different IP algorithms on the Timik dataset with the default parameters ( $(k, m, n) = (50, 10000, 125)$ ). For every MIP algorithm, we evaluate the solution quality of different algorithms with the running time constrained by 200, 1000 and 5000 times the running time of our proposed AVG-D algorithm for the same instance to compare the obtained solutions in different time limits. The y-axis shows the objective value normalized by the solution of AVG-D, which manifests none of the 5 baselines achieves any solution better than that of AVG-D in 5000X of the running time of AVG-D. As such, although there is some subtle difference in performance across different MIP algorithms, none of the examined algorithms shows a reasonable scalability.

Figure 9(b) shows the effects of the speedup strategies on AVG and AVG-D. AVG-ALP and AVG-D-ALP are AVG and AVG-D without the advanced LP transformation technique, respectively, while AVG-AS and AVG-D-AS are AVG and AVG-D without the advanced focal parameter sampling technique, respectively. The results manifest that all speedup strategies effectively boost the efficiency of AVG and AVG-D. For AVG, the advanced LP transformation

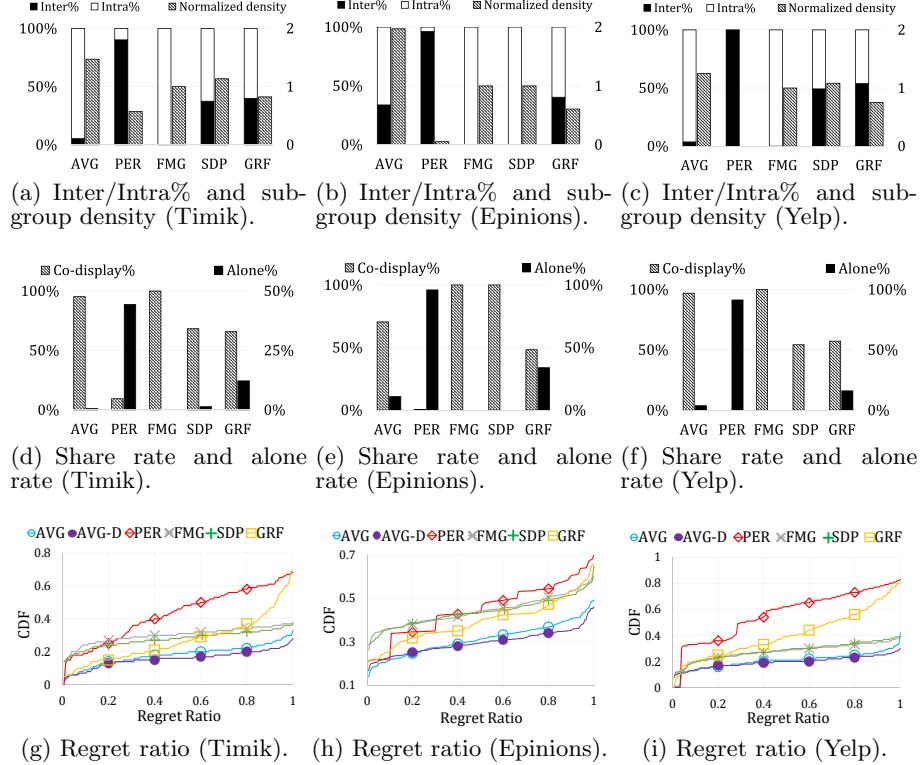
plays a more dominant role than the advanced focal parameter sampling scheme since the computational cost of solving the linear program is more evident in AVG. Conversely, as AVG-D needs to carefully examine every combination of *good* focal parameters (i.e., those sets of parameters that do assign an item to at least one user), the advanced sampling technique has a significant effect in filtering out bad focal parameters. Therefore, the effect of the advanced LP transformation pales in comparison to that of the advanced sampling technique in AVG-D.

## 6.5 Comparisons on Subgroup Metrics

In the following, we analyze the subgroups in the SAVG  $k$ -Configurations returned from all algorithms<sup>5</sup> in terms of subgroup-related metrics. Figures 10(a), 10(b), and 10(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 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 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.

<sup>5</sup>Some of the results of AVG-D are omitted as they are similar to AVG.



**Figure 10: Comparisons on subgroup metrics.**

Figures 10(d), 10(f), and 10(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 10(g), 10(i), and 10(h) report the CDF of regret ratios of all algorithms in the Timik, Yelp, and Epinions datasets. The regret ratio  $\text{reg}(u)$  [46] is a game-theory based measurement for satisfaction of individual users and the overall *fairness* of the solution. For each user  $u$ , the regret ratio  $\text{reg}(u)$ , and its converse, *happiness* ratio  $\text{hap}(u)$ , are defined as

$$\text{reg}(u) \equiv 1 - \text{hap}(u), \quad \text{hap}(u) \equiv \frac{\sum_{c \in \mathbf{A}^*(u, \cdot)} w_{\mathbf{A}}(u, c)}{\max_{C_u} \sum_{c \in C_u} \bar{w}_{\mathbf{A}}(u, c)}$$

where the numerator in  $\text{hap}(u)$  is the achieved SAVG utility, the denominator with  $\bar{w}_{\mathbf{A}}(u, c) = (1 - \lambda) \cdot p(u, c) + \lambda \cdot \sum_{v \in V} \tau(u, v, c)$  is an *upper bound* of possible SAVG utility when all users view  $c$  together with user  $u$ , and  $C_u$  is a  $k$ -itemset, corresponding to a very optimistic scenario favoring  $u$  the most. Note that the second term of  $\bar{w}_{\mathbf{A}}(u, c)$  is different from  $w_{\mathbf{A}}(u, c)$  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 high  $\text{hap}(u)$  (equivalently, a low  $\text{reg}(u)$ ) 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. 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.

## 6.6 A Case Study on Partitioning Subgroups

Figure 11 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.

## 6.7 Sensitivity Test of AVG-D

In the following, we compare AVG-D with the optimal solution in the Timik dataset. Figure 12(a) shows that with  $r$  values from 0.7 to 1.0, AVG-D finds nearly-optimal solutions as Co-display Subgroup Formation with  $r$  balances between the current increment in SAVG utility and the potential SAVG utility in the future. This phenomenon is also observed in other datasets. However, setting  $r = 0.25$  is still promising as it achieves 86.1% of the optimal objective, which is consistent with the theoretical guarantee. Figure 12(a) also manifests that AVG-D with small values of  $r$  tends to resemble the group approach in Section 1 (i.e., displaying the same items to every user). For instance, AVG-D finds a large subgroup of all users with  $r \leq 0.2$ . AVG-D gradually becomes more alike to the personalized approach as  $r$  grows (e.g., the social utility is close to 0 for  $r \geq 1.5$ , implying almost every user is displayed her own favorite items). This is actually consistent with the design of AVG-D, since 1) a small  $r$  essentially ignores the potential utility gain in the future from allocating other items than the current focal item, leading to a greedy behavior of always including every eligible user into the targeted subgroup in CSF. Consequently, running every iteration with  $r$  close to 0 finds a

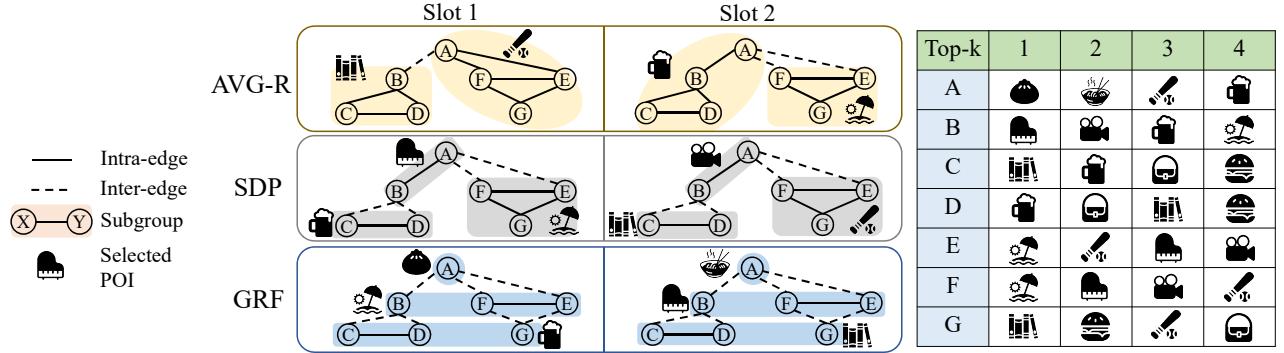
large subgroup of all users in every slot; 2) a large  $r$  prioritizes only the future gain and thereby tends to select very few users into the targeted subgroup. This reluctant behavior in the extreme (consider  $r = \infty$ ) thus leads to subgroups consisting of only one user, essentially simulating the personalized approach. In summary, this again manifests that AVG-D can strike a good balance between the personalized and group approaches.

Figure 12(b) depicts the execution time of AVG-D with different values of  $r$ . Intuitively, when AVG-D chooses fewer users in the current targeted subgroup, more iterations are required to construct a complete SAVG  $k$ -Configuration. For  $n$  users and  $k$  slots, the total number of iterations is  $k$  if every targeted subgroup includes all  $n$  users but would increase to  $kn$  if every iteration displays the focal item to only one user. Therefore, AVG-D has a larger execution time with larger values of  $r$ . Figures 12(c) and 12(d) further show the average normalized density and Inter/Intra% among the subgroups in the solutions of AVG-D with different values of  $r$ . Consistent with the above discussion, the behavior of AVG-D essentially spans a spectrum from the personalized to the group approach.

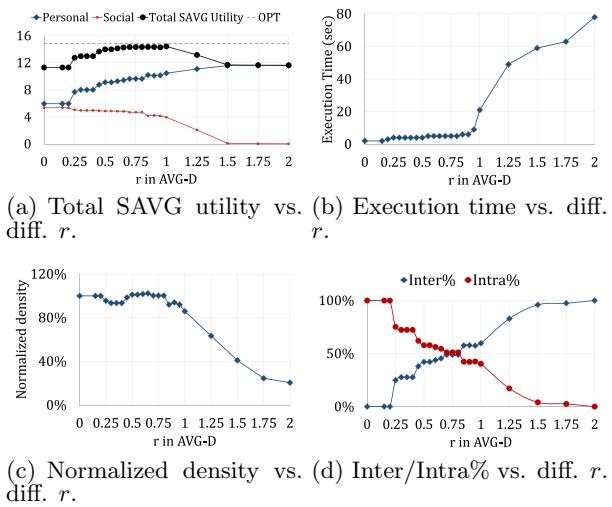
## 6.8 Experimental Results for SVGIC-ST

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 13(a) and 13(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.

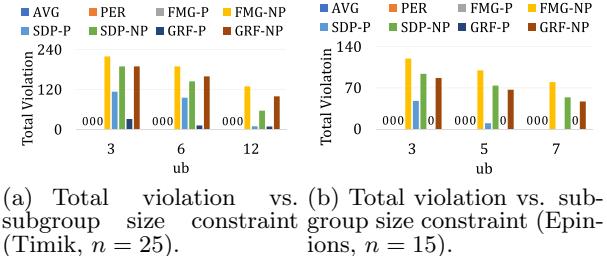
Figures 14 and 15 compares the total SAVG utility achieved by all methods *with pre-partitioning* in Timik and Epinions, respectively, with  $n = 15$  and the subgroup size constraint varies from 3 to 15, where infeasible solutions achieves a total SAVG utility of 0. Note that all the baseline methods, 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 Epin-



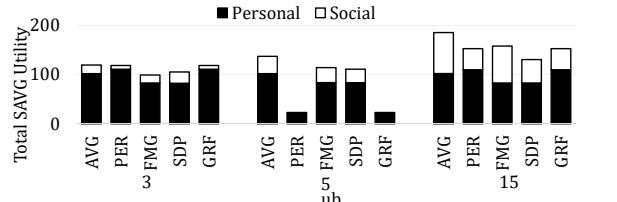
**Figure 11: Illustration of subgroup partitioning approaches.**



**Figure 12: Sensitivity test on  $r$  in AVG-D.**

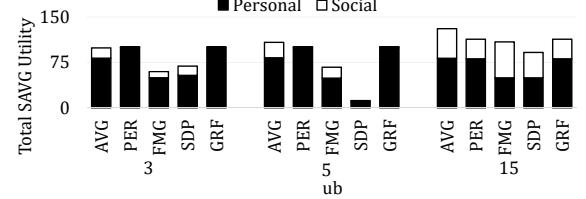


**Figure 13: Comparisons on SVGIC-ST.**



**Figure 14: Total SAVG utility vs. subgroup size constraint (Timik,  $n = 15$ ).**

Top-k	1	2	3	4
A	Bucket	Bowl	Pencil	Cup
B	Cup	Movie Camera	Cup	Swimmer
C	Books	Movie Camera	Bucket	Hamburger
D	Cup	Bucket	Books	Hamburger
E	Swimmer	Pencil	Bucket	Movie Camera
F	Swimmer	Movie Camera	Movie Camera	Pencil
G	Books	Hamburger	Pencil	Cup



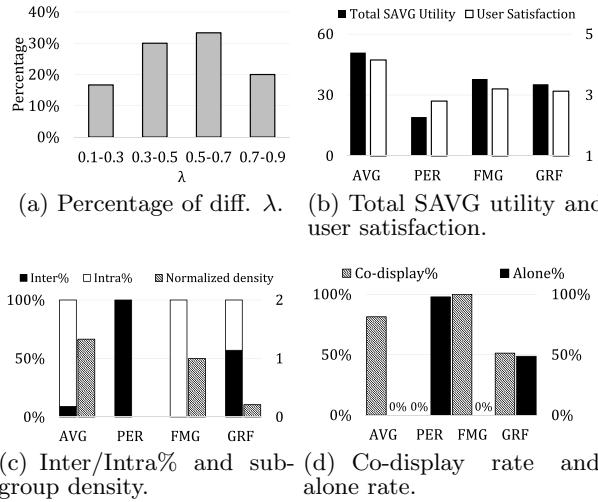
**Figure 15: Total SAVG utility vs. subgroup size constraint (Epinions,  $n = 15$ ).**

ions). 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 feasibility 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.

## 6.9 User Study

For the user study, we recruit 44 participants to visit our VR store. Their social networks, preference utility, and  $\lambda$  are pre-collected with questionnaires, which follows the setting of [65], and the social utility is learned by PIERT [47]. A Likert scale questionnaire [79] is used to find the preference utility of items, where users are allowed to discuss the products so that the social utility can be learned. Finally, they are asked to provide  $\lambda$  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 [79] 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 16(a) reports that  $\lambda$  values specified by the users range from 0.15 to 0.85 with the average as 0.53, indicating



**Figure 16: Comparisons on user study.**

that both personal preferences and social interactions are essential in VR group shopping. Figure 16(b) compares the total SAVG utility as well as the recorded user satisfaction 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 ( $p\text{-value} \leq 0.019 < 0.05$ ). It is worth noting that the correlation between the SAVG utility and user satisfaction is high (Spearman correlation 0.835; Pearson correlation 0.814), which manifest that the SAVG utility is a good estimation of user satisfaction.

Figures 16(c) and 16(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 10), 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%.

## 7. 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 SVGIC is NP-hard to approximate within  $\frac{32}{31} - \epsilon$ . 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.

## 8. REFERENCES

- [1] 3D virtual stores - the new virtual reality social shopping in 3D. <https://www.qbittech.com/index.php/component/k2/item/41-virtual-reality-ecommerce-3d-stores>. Accessed: 2020-02-03.
- [2] Alibaba offers VR shopping. <https://edition.cnn.com/videos/world/2016/11/28/alibaba-vr-shopping-stevens-pkg.cnn>. Accessed: 2019-09-30.
- [3] AltspaceVR. <https://altrvr.com/>. Accessed: 2020-02-03.
- [4] F. Amato, V. Moscato, A. Picariello, and G. Sperli. Influence maximization in social media networks using hypergraphs. In *GPC*, pages 207–221, 2017.
- [5] A. An, M. Kargar, and M. Zihayat. Finding affordable and collaborative teams from a network of experts. In *SDM*, pages 587–595. SIAM, 2013.
- [6] E. Bozgeyikli, A. Raji, S. Katkoori, and R. V. Dubey. Point & teleport locomotion technique for virtual reality. In *CHI PLAY*, pages 205–216. ACM, 2016.
- [7] C. Cai, R. He, and J. J. McAuley. SPMC: socially-aware personalized Markov chains for sparse sequential recommendation. In *IJCAI*, pages 1476–1482, 2017.
- [8] Can virtual experiences replace reality? The future role for humans in delivering customer experience. [https://www.oracle.com/webfolder/s/delivery\\_production/docs/FY16h1/doc35/CXResearchVirtualExperiences.pdf](https://www.oracle.com/webfolder/s/delivery_production/docs/FY16h1/doc35/CXResearchVirtualExperiences.pdf). Accessed: 2020-02-03.
- [9] D. Cao, X. He, L. Miao, Y. An, C. Yang, and R. Hong. Attentive group recommendation. In *SIGIR*, pages 645–654. ACM, 2018.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] L. Chen, C. Liu, R. Zhou, J. Li, X. Yang, and B. Wang. Maximum co-located community search in large scale social networks. *VLDB*, 11(10):1233–1246, 2018.
- [14] Y. Cheng, Y. Yuan, L. Chen, C. G. Giraud-Carrier, and G. Wang. Complex event-participant planning and its incremental variant. In *ICDE*, pages 859–870. IEEE, 2017.
- [15] M. B. Cohen, Y. T. Lee, and Z. Song. Solving linear programs in the current matrix multiplication time. In *FOCS*. IEEE, 2019.
- [16] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *Journal on Computing*, 23(4):864–894, 1994.
- [17] Douban Location. <http://www.douban.com/location/>. Accessed:

- 2020-02-03.
- [18] X. Dreze, S. J. Hoch, and M. E. Purk. Shelf management and space elasticity. *Journal of retailing*, 70(4):301–326, 1994.
- [19] A. Ene, J. Vondrák, and Y. Wu. Local distribution and the symmetry gap: Approximability of multiway partitioning problems. In *SODA*, pages 306–325. SIAM, 2013.
- [20] Epinions dataset. [http://www.trustlet.org/extended\\_epinions.html](http://www.trustlet.org/extended_epinions.html).
- [21] Facebook Events. <https://www.facebook.com/events/>. Accessed: 2020-02-03.
- [22] Facebook Horizon. <https://www.oculus.com/facebookhorizon/>. Accessed: 2020-02-03.
- [23] Facebook social VR demo - Oculus connect 2016. <https://youtu.be/YuIgyKLPt3s>. Accessed: 2019-09-30.
- [24] From virtual reality to personalized experiences: Alibaba is bringing us the future of retail this singles day. <https://www.forbes.com/sites/helenwang/2016/11/06/how-alibaba-will-use-the-worlds-biggest-shopping-day-to-transform-retail/#41d6340c6d4e>. Accessed: 2020-02-03.
- [25] Future of retail: Artificial intelligence and virtual reality have big roles to play. <https://www.forbes.com/sites/rachelarthur/2016/06/15/future-of-retail-artificial-intelligence-and-virtual-reality-have-big-roles-to-play/#b87b8ae7f9df>. Accessed: 2020-02-03.
- [26] 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.
- [27] 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.
- [28] G. Guo, J. Zhang, and N. Yorke-Smith. TrustSVD: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In *AAAI*, 2015.
- [29] Gurobi optimizer. <http://www.gurobi.com>. Accessed: 2019-09-30.
- [30] J. Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001.
- [31] R. He and J. J. McAuley. VBPR: visual Bayesian personalized ranking from implicit feedback. In *AAAI*, pages 144–150. AAAI, 2016.
- [32] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua. Neural collaborative filtering. In *WWW*, pages 173–182. ACM, 2017.
- [33] 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.
- [34] How will V-Commerce arrive? consumers show the way. <https://www.lek.com/insights/ei/how-will-v-commerce-arrive-consumers-show-way>. Accessed: 2020-02-03.
- [35] 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, 2014.
- [36] H.-J. Hung et al. When social influence meets item inference. In *KDD*. ACM, 2016.
- [37] IBM CPLEX optimizer. <https://www.ibm.com/analytics/cplex-optimizer>. Accessed: 2019-09-30.
- [38] IKEA VR experience. [https://store.steampowered.com/app/447270/IKEA\\_VR\\_Experience/](https://store.steampowered.com/app/447270/IKEA_VR_Experience/). Accessed: 2020-02-03.
- [39] IrisVR. <https://irisvr.com/>. Accessed: 2019-09-30.
- [40] 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.
- [41] 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.
- [42] S. Ko, H. Lai, H. Shuai, D. Yang, W. Lee, and P. S. Yu. Optimizing item and subgroup configurations for social-aware VR shopping (full version). *CoRR*, abs/2002.04338, 2020.
- [43] M. Langberg, Y. Rabani, and C. Swamy. Approximation algorithms for graph homomorphism problems. In *APPROX-RANDOM*, volume 4110, pages 176–187, 2006.
- [44] 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.
- [45] 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.
- [46] K. Li, W. Lu, S. Bhagat, L. V. S. Lakshmanan, and C. Yu. On social event organization. In *KDD*, pages 1206–1215. ACM, 2014.
- [47] 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.
- [48] R. Lissermann, J. Huber, M. Schmitz, J. Steimle, and M. Mühlhäuser. Permutin: mixed-focus collaboration on multi-view tabletops. In *CHI*, pages 3191–3200. ACM, 2014.
- [49] Lowes Innovation Labs. The next-generation Lowes holoroom. <https://youtu.be/DVsEb9vla-I>. Accessed: 2020-02-03.
- [50] 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.
- [51] Z. Lu, Z. Dou, J. Lian, X. Xie, and Q. Yang. Content-based collaborative filtering for news topic recommendation. In *AAAI*, pages 217–223, 2015.
- [52] X. Luo. How does shopping with others influence impulsive purchasing? *Journal of Consumer Psychology*, 15(4):288–294, 2005.

- [53] Macys ramps up tech push with virtual reality and personalized online shopping.  
<https://fortune.com/2018/03/19/macys-tech-2/>. Accessed: 2020-02-03.
- [54] R. Manokaran, J. Naor, P. Raghavendra, and R. Schwartz. SDP gaps and UGC hardness for multiway cut, 0-extension, and metric labeling. In *STOC*, pages 11–20. ACM, 2008.
- [55] P. Manurangsi. Almost-polynomial ratio eth-hardness of approximating densest k-subgraph. In *STOC*, pages 954–961. ACM, 2017.
- [56] Meetup. <https://www.meetup.com/>. Accessed: 2020-02-03.
- [57] 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.
- [58] New Walker Sands study: Drones and virtual reality as a solution for the modern retail experience.  
<https://www.walkersands.com/new-walker-sands-study-drones-and-virtual-reality-as-a-solution-for-the-modern-retail-experience/>. Accessed: 2020-02-03.
- [59] Qbit. <https://www.qbittech.com/index.php>. Accessed: 2020-02-03.
- [60] S. Qi, N. Mamoulis, E. Pitoura, and P. Tsaparas. Recommending packages to groups. In *ICDM*, pages 449–458. IEEE, 2016.
- [61] 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.
- [62] Rec Room. <https://recroom.com/rec-room>. Accessed: 2020-02-03.
- [63] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461, 2009.
- [64] Retailers have a lot to gain from AR and VR.  
<https://www.forbes.com/sites/cognitiveworld/2019/10/01/retailers-have-a-lot-to-gain-from-ar-and-vr/#4b4278577a1c>. Accessed: 2020-02-03.
- [65] S. B. Roy, L. V. S. Lakshmanan, and R. Liu. From group recommendations to group formation. In *SIGMOD*, pages 1603–1616. ACM, 2015.
- [66] 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.
- [67] D. Serbos, S. Qi, N. Mamoulis, E. Pitoura, and P. Tsaparas. Fairness in package-to-group recommendations. In *WWW*, pages 371–379. ACM, 2017.
- [68] A. Sharma and J. Vondrák. Multiway cut, pairwise realizable distributions, and descending thresholds. In *STOC*, pages 724–733. ACM, 2014.
- [69] J. She, Y. Tong, and L. Chen. Utility-aware social event-participant planning. In *SIGMOD*, pages 1629–1643. ACM, 2015.
- [70] J. She, Y. Tong, L. Chen, and C. C. Cao. Conflict-aware event-participant arrangement and its variant for online setting. *TKDE*, 28(9):2281–2295, 2016.
- [71] 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, 2018.
- [72] 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.
- [73] C. Shen, D. Yang, W. Lee, and M. Chen. Spatial-proximity optimization for rapid task group deployment. *ACM TKDD*, 10(4):47:1–47:36, 2016.
- [74] Shopify. <https://www.shopify.com/>. Accessed: 2020-02-03.
- [75] 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.
- [76] H. Shuai, D. Yang, P. S. Yu, and M. Chen. Willingness optimization for social group activity. *PVLDB*, 7(4):253–264, 2013.
- [77] V. Sigurdsson et al. Brand placement and consumer choice: an in-store experiment. *Journal of Applied Behavior Analysis*, 2009.
- [78] The world’s first virtual reality department store - brought to you by eBay and Myer.  
<https://youtu.be/yAuiXhJPnr8>. Accessed: 2019-09-30.
- [79] W. M. Trochim and J. P. Donnelly. *Research methods knowledge base*, volume 2. Atomic Dog Publishing Cincinnati, OH, 2001.
- [80] VentureBeat: Amazon virtual reality VR mall kiosks - the future of retail.  
<https://youtu.be/J5NviNVd0sc>. Accessed: 2019-09-30.
- [81] Virtual reality store IKEA.  
[https://youtu.be/5\\_znFPj5Lis](https://youtu.be/5_znFPj5Lis). Accessed: 2019-09-30.
- [82] Virtual reality: Tourism firms use VR to attract visitors. <https://bbc.in/2MExzzx>. Accessed: 2019-09-30.
- [83] VRChat. <https://vrchat.net/>. Accessed: 2019-09-30.
- [84] VRChat guides. <https://vrchat.com/guides>. Accessed: 2020-02-03.
- [85] 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.
- [86] D. P. Williamson and D. B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011.
- [87] Worldwide augmented and virtual reality spending guide.  
<https://www.idc.com/getdoc.jsp?containerId=prUS45679219>. Accessed: 2020-02-03.
- [88] Y. Xu, R. Goebel, and G. Lin. Submodular and supermodular multi-labeling, and vertex happiness. *CoRR*, abs/1606.03185, 2016.
- [89] B. Yang, Y. Lei, D. Liu, and J. Liu. Social collaborative filtering by trust. In *IJCAI*, 2013.

- [90] Yelp dataset. <https://www.yelp.com/dataset>. Accessed: 2019-09-30.
- [91] 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.
- [92] P. Zhang and A. Li. Algorithmic aspects of homophily of networks. *Theoretical Computer Science*, 593:117–131, 2015.
- [93] P. Zhang, Y. Xu, T. Jiang, A. Li, G. Lin, and E. Miyano. Improved approximation algorithms for the maximum happy vertices and edges problems. *Algorithmica*, 80(5):1412–1438, 2018.
- [94] 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.
- [95] 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.
- [96] L. Zhao, H. Nagamochi, and T. Ibaraki. Greedy splitting algorithms for approximating multiway partition problems. *Mathematical Programming*, 102(1):167–183, 2005.
- [97] 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.
- [98] 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.
- [99] 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 TKDE*, 28(5):1147–1159, 2016.
- [100] J. Zhu, J. Zhu, S. Ghosh, W. Wu, and J. Yuan. Social influence maximization in hypergraph in social networks. *IEEE Transactions on Network Science and Engineering*, 6(4):801–811, 2019.