Supplementary information: Leveraging Monte Carlo Tree Search for Group Recommendation

Antonela Tommasel
ISISTAN, CONICET-UNCPBA
Tandil, Argentina
antonela.tommasel@isistan.unicen.edu.ar

J. Andres Diaz-Pace
ISISTAN, CONICET-UNCPBA
Tandil, Argentina
andres.diazpace@isistan.unicen.edu.ar

ABSTRACT

This document includes additional methodology details, Tables with results and charts. The companion repository can be found at: https://github.com/tommantonela/MCTS_group_recommendation.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

recommender systems, group recommendation, Large Language Models, heuristic search, novelty, diversity, fairness

1 TASK AND METHOD

Reward policy configuration. To explore the role of LLMs in assessing rewards [5, 15], we defined strategies aiming at analyzing the interactions between user and group preferences and candidate items. We replaced the MCTS strategies by an evaluation prompt, structured as follows:

1) **Group history**. A set of items liked by all members of the group, providing insight into common group interests.

2) **User individual history**. For each user, a set of items exclusively liked by them (i.e., items were either disliked or not rated by the other group members), showcasing their specific interests.

3) **Recommended items**. The sequence of items in the path from the root node to the current node.

4) **Task to solve**. The LLM was instructed to return a numeric score (between 0 and 100) based on different strategies.

The system prompt for the LLM was defined as follows:

> *You are a helpful movie recommender system. Your task is to recommend movies to a group of people based on their watching history.*
> *You will receive:*
> - The group preferences.
> - The individual user preferences.
> - A list of movies recommended to the group.
>
> *Your task is to use all the provided information to generate a score for the recommended movies. You have access to all the information you need.*

This is the prompt describing the task used for the evaluation.

> *These are the "group preferences":* `{group_history}` (1)
>
> *These are the "individual user preferences":* `{users_history}` (2)
>
> *Recommended movies:* `{recommended_movies}` (3)
>
> *Your task is:* `{task_to_solve}` (4)
>
> *All the information you need is available in this conversation, focus on the "group preferences", the "individual user preferences" and the "recommended movies". Do not include anything else, except your numeric score, in your answer.*
>
> *Your JSON answer:*

Then, the `task_to_solve` was defined according to the following strategies[1]:

i) ***Base (MCTS-LLM)***: The LLM is asked to evaluate candidate items based on group preferences and individual preferences, considering how well movies satisfy the whole group focusing on diversity and novelty.

> 1. Using the "*group preferences*" and the "*individual user preferences*", generate a numeric score for each of the "*recommended movies*" based on how well they satisfy the group as a whole. The score should range between 0 and 100, where 0 means no satisfaction and 100 means full satisfaction.
>
> 2. In your assessment also consider the degrees of diversity and novelty of each of the "*recommended movies*" with respect to the "*group preferences*" and "*individual group preferences*". The higher the diversity or the novelty, the better.

As another variant, we indicated that the score should also consider the order of candidate items by adding the instruction below:

> 3. The order of the recommended movies is important. With the scores computed by movie, compute a score for the whole list, considering the individual scores for each movie, their position in the ranking, and the relationship between a movie and the ones ranked before.

ii) ***Ranking (MCTS-LLM-rank)***: The LLM is asked to evaluate candidate items based on group preferences, individual preferences, and the recommendations made for each group member. The score should reflect diversity, novelty, fairness, or their combination. Unlike *MCTS-LLM*, the LLM is informed about which items were recommended to each user, even if those movies are not part of the candidate set. The prompt is modified to also include "*the list of movies recommended to each member of the group*".

[1]Each strategy included additional instructions regarding what to include in the JSON answer, which can be found in the companion repository.

1. Using the "*group preferences*", the "*individual user preferences*" and the "*recommendations for each group member*", generate a numeric score for each of the "*group recommended movies*" based on how well they satisfy the group as a whole. The score should range between 0 and 100, where 0 means no satisfaction and 100 means full satisfaction.

2. The score should reflect the degrees of `diversity - novelty - fairness - diversity, novelty and fairness` of each of the "*group recommended movies*" with respect to the "*group preferences*", "i*ndividual group preferences*" and "*recommendations for each group member*". The higher the diversity or the novelty, the better.

3. The order of the "*recommendations for each group member*" and "*group recommended movies*" is important. With the scores computed by movie, compute a score for the whole list, considering the individual scores for each movie, their position in the ranking, and the relationship between a movie and the ones ranked before.

iii) ***Position (MCTS-LLM-pos)***: Similar to *MCTS-LLM-rank*, but instead of the recommendations for each user, the LLM receives information of whether the items were recommended to each user and in which ranking positions. The prompt is modified to also include: "*The position in which each movie was recommended to each user. A -1 indicates that the movie was not recommended to the user.*".

1. Using the "*group preferences*", the "*individual user preferences*" and the "*positions in which each movie was recommended to the user*", generate a numeric score for each of the "group recommended movies" based on how well they satisfy the group as a whole. The score should range between 0 and 100, where 0 means no satisfaction and 100 means full satisfaction. Generate a score for each movie in the list.

2. The score should reflect the degrees of `diversity - novelty - fairness - diversity, novelty and fairness` of each of the "*group recommended movies*" with respect to the "*group preferences*", "*individual group preferences*" and "*positions in which each movie was recommended to the user*". The higher the diversity or the novelty, the better.

3. The order of the "*group recommended movies*" and in which positions they were recommended to the users is important. With the scores computed by movie, compute a score for the whole list, considering the individual scores for each movie, their position in the ranking, and the relationship between a movie and the ones ranked before.

2 EXPERIMENTAL SETUP

Data collection. Our evaluation used the *MovieLens* dataset[2], including user ids, movies, ratings, genres, and timestamps. No group information was originally included in the dataset. To account for shifts in user interests due to extensive watching histories, we only considered ratings created after January 1, 2016. We kept users that rated more than 30 movies, resulting in over 29k users and over 56k movies. Assuming that the most recently watched movies better reflect current interests, enabling more timely recommendations, each users' ratings were split based on timestamps, with the first 60% ratings as training, the next 20% as validation and the remaining 20% as testing. The test set of each group member only includes movies that were not part of any member train to avoid recommending seen elements again as part of the group recommendation [2]. Movies were considered liked if rated 4 or higher.

Group formation. Although in previous literature (e.g., [1, 13]), analyses mostly revolved around similar groups, to explore how fairness varies according to group size and the homogeneity/divergence of interests, in addition to the strategy described and reported in the main document, we adopted two strategies from prior research [2, 14] to create groups of 2 to 8 members:

i) **random**: simulating groups with unrelated members, we randomly selected users without replacement;

ii) **divergent**: similar to the similar group creation strategy, but with one user being intentionally dissimilar to the rest.

We generated 150 groups of each type. The average group similarity was 0.14, 0.23 and 0.15 for random, similar, and divergent groups, respectively. On average, every pair of users in the similar groups, rated a minimum of 9 (median 22) common movies, which, according to [1] helps to ensure the reliability of the similarity.

LLM choice. The LLM-based strategies used `Gemma:2b-instruct`, which showed a good balance between execution time and answer quality (in terms of prompt adherence). We also evaluated `Mistral:7b` and `Llama3`, but discarded them due to either excessive execution times on the available hardware or inappropriate prompt adherence, such as including Python code in the responses. Finally, given the exploratory nature of the work, we discarded `GPT` due to costs. We plan to expand the set of used LLMs in future works.

Baselines. We compared the performance of the proposed aggregation technique with different approaches: i) Commonly-used aggregation strategies based on social choice [4]: *additive*, *least-misery*, and *Borda* rank aggregation; ii) *GreedyLM* [14], a greedy technique using Least Misery Fairness; iii) *GFAR* [2][3] ; iv) *EP-FuzzDA* [3][4] ; and v) LLM-based aggregation (*LLM-agg*). The LLM was provided with groups' common watching history, users' individual watching history, the list of movies recommended to each user, and the task to solve. The LLM was asked to consider the provided information and, aggregate the individual recommended lists into one that satisfying all group members.

Evaluation metrics. Evaluating recommenders solely on the relevance of recommendations offers a partial view of their performance [8]. Therefore, we considered both relevance and different beyond-accuracy perspectives, namely: coverage, ordering and diversity/novelty metrics. For *relevance*, we selected precision and nDCG[5]. For *coverage*, we used *zero-recall* (as used in [2]), which measures the fraction of group members who did not receive any relevant recommendation. This metric can be also considered a fairness metric, as values close to zero indicate that every group

[2]https://grouplens.org/datasets/movielens/25m/

[3]https://github.com/mesutkaya/recsys2020

[4]https://github.com/LadislavMalecek/UMAP2021

[5]We followed the binary definition, which defines items as relevant or non-relevant.

	precision (↑)	nDCG (↑)	zrecall (↓)	corr	diversity (↑)	novelty (↑)	novelty$_{min-max}$ (↑)
LLM-agg	0.127±0.054	0.273±0.098	0.584±0.152	0.199	0.721±0.156	0.746±0.034	0.921
additive	0.099±0.035	0.238±0.073	0.623±0.11	0.075	0.640±0.071	0.720±0.023	0.914
least-misery	0.127±0.068	0.259±0.117	0.641±0.182	**0.480**	0.475±0.259	0.717±0.049	0.892
Borda	0.120±0.057	0.291±0.123	0.536±0.196	0.147	0.709±0.078	0.732±0.031	0.926
GreedyLM [14]	0.100±0.066	0.244±0.151	0.610±0.240	0.192	0.744±0.087	0.748±0.029	*0.932*
GFAR [2]	0.114±0.055	0.288±0.121	0.536±0.189	-0.083	0.761±0.083	0.751±0.029	*0.932*
EP-FuzzDA [3]	0.117±0.056	0.285±0.126	0.546±0.199	0.222	0.747±0.082	0.746±0.031	*0.932*
MCTS-diversity	0.105±0.053	0.263±0.118	0.576±0.189	0.239	**0.872**±0.027	0.779±0.022	**0.965**
MCTS-novelty	0.125±0.055	0.295±0.111	0.535±0.174	0.330	0.737±0.104	**0.806**±0.023	*0.936*
MCTS-MRR	0.108±0.054	0.269±0.122	0.568±0.196	0.253	*0.867*±0.031	*0.790*±0.023	**0.961**
MCTS-fairness	0.105±0.049	0.262±0.109	0.578±0.177	0.359	0.676±0.095	0.722±0.039	*0.931*
MCTS-weighted	0.120±0.057	0.287±0.119	0.546±0.189	0.292	0.742±0.112	**0.809**±0.023	*0.937*
MCTS-LLM	*0.140*±0.061	*0.317*±0.123	*0.508*±0.192	0.352	0.666±0.081	0.710±0.031	0.927
MCTS-LLM-rank	0.128±0.039	0.285±0.042	0.560±0.080	*0.440*	0.681±0.049	0.724±0.014	*0.939*
MCTS-LLM-pos	**0.148**±0.031	**0.344**±0.071	**0.460**±0.128	0.340	0.700±0.067	0.735±0.034	0.941

Table 1: Summarized mean group scores and standard deviations ($k = 5$) (best results in bold, second best in *italic*)

member received at least one relevant item. As users watch movies in a certain order, and *order* is important for generating the sequences of candidate movies (by the MCTS algorithm), we considered Kendall's Tau rank correlation between the actual correlation order and the recommendation sequence[6]. For *diversity* and *novelty*, we followed the same definitions as in the reward strategies. Enhancing diversity/novelty is also associated with increased fairness and reduced biases [9], as they help broadening users' social experiences [9].

Eq. 1 and Eq. 2 present the definitions for diversity and novelty, respectively where R represents the group recommendations, M_u the items with whom user u has already interacted and $d_m(i, j)$ the dissimilarities between items i and j.

$$Diversity(u) = \frac{1}{|R|(|R|-1)} \sum_{i \in R} \sum_{j \in R} d_m(i, j) \tag{1}$$

$$Novelty(u) = \frac{1}{|R||M_u|} \sum_{i \in R} \sum_{j \in M_u} d_m(i, j) \tag{2}$$

A cut-off threshold was set to select the top-K recommended items, with $K = 5$. Metrics were computed at the individual user level, and then aggregated to obtain the group score. For each user, recommendations were considered correct if they appeared in their test set. Group-level results were summarized from user results using metrics such as minimum, mean, median, variance, and maximum scores [2, 6, 7, 10]. To assess the statistical significance of differences, the Wilcoxon paired samples test ($\alpha = 0.05$) was used, including corrections for multiple tests.

3 ANALYSIS

Table 1 presents the mean (and corresponding standard deviation) results for groups of 5 members, created using the similar group strategy. The metrics for each group were computed as the mean score for each group member. In the case of correlations, a meta-analysis [11, 12] was performed to compute their mean score. Inspired by [14] we also included the min-max ratio for novelty. In the case of precision and nDCG as the minimum was 0 in most cases, the resulting min-max ratio was also 0.

The MTCS algorithm was run for 200 iterations, and no simulation strategy was used. Due to the random nature of some MTCS strategies, we ran the evaluations multiple times using different seeds.

Figure 1 and Figure 2 show the distribution of mean and max scores for the similar groups with 5 members. As the Figures show, similar tendencies are observed in both summarizations[7].

Finally, Table 2 indicates whether the observed differences were statistically significant.

3.1 RQ1. Baselines versus MCTS

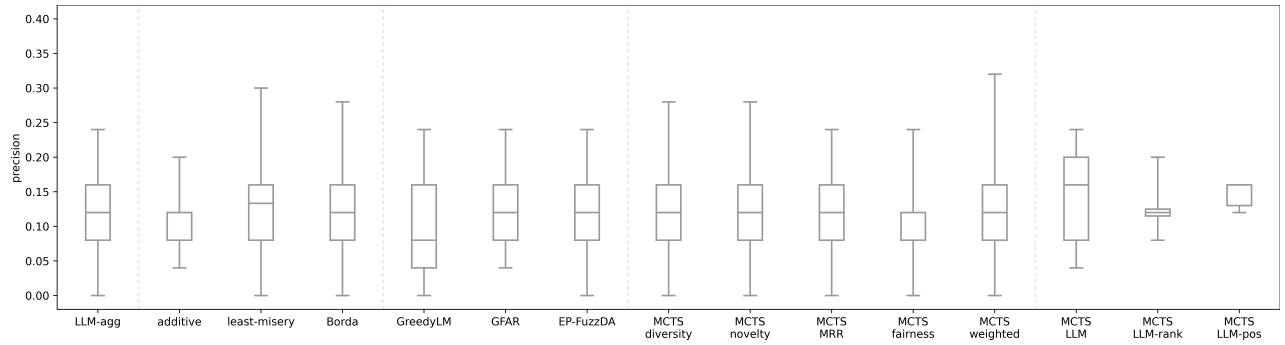
For different group sizes, the lowest recommendation relevance was observed for the largest groups, while the highest relevance was found in the smallest groups. This pattern aligns with other studies Kaya et al. [2]. The smallest differences among strategies were also seen for the smallest groups, with no alternative showing a distinctly higher performance. For groups of 4 and 8 members, the MCTS variants showed significant differences, with a few exceptions in precision and correlation. Similar trends were observed for divergent and random strategies for group creation with respect to the similar group strategy. Overall, scores were slightly higher for divergent and random groups, with the most significant differences seen in the divergent strategy.

3.2 Exploratory analysis of simulation strategies and number of iterations

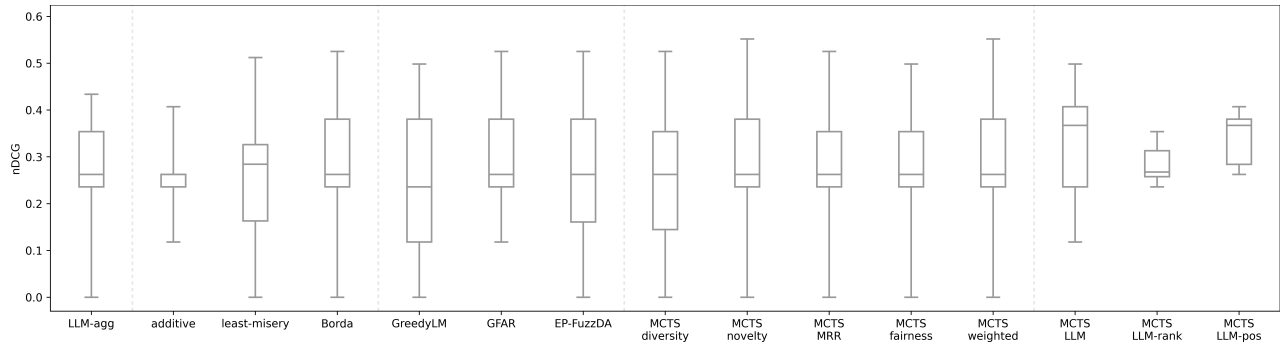
We evaluated increasing the number of iterations from 200 to 500, 1000 and 1500. Although in general, more iterations slightly modified the results, differences were not significant. The only exceptions were for *MCTS-diversity*, for which using 200 iterations achieved significantly better precision and nDCG (with small effect sizes), while increasing the iterations allowed to significantly boost the

[6] As correlation metrics require sequences to at least have the same number of elements, from the recommended movies, the analysis only included the sub-set of correctly recommended ones.

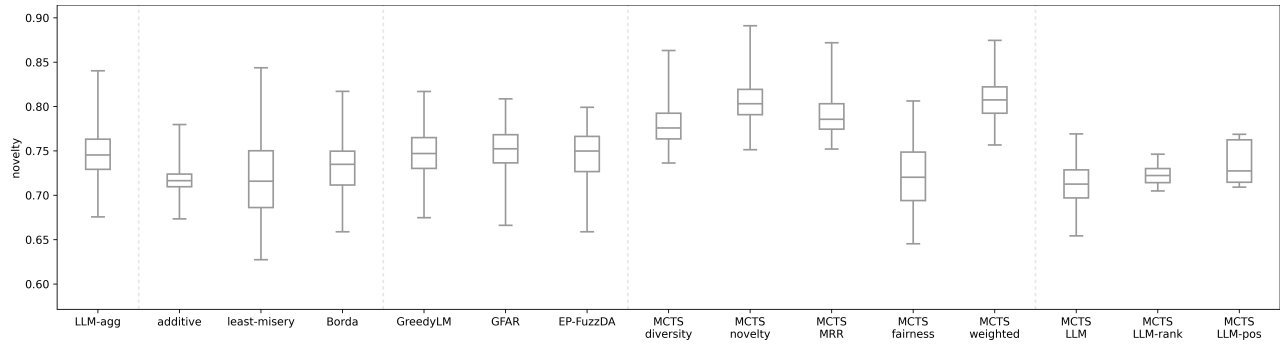
[7] Results for the min and variance summarizations can be found in the companion repository.



(a) Mean precision distribution across groups



(b) Mean nDCG distribution across groups



(c) Mean novelty distribution across groups

Figure 1: Mean metric distribution across groups (similar groups of 5 members)

diversity and novelty of recommendations (with medium to large effect sizes). For divergent, similar tendencies were observed, with the exception that *MCTS-novelty* had its diversity/novelty increased.

We also evaluated introducing a random simulation strategy by which the last 2 elements in the recommendation sequence (of size 5) were simulated. In general, the observed differences were not statistically significant. There were only a few exceptions. For example, introducing simulations allowed *MCTS-diversity* to increase nDCG (with a small effect size), while it also caused a decrement of novelty (*MCTS-weighted* and *MCTS-MRR*) and precision (*MCTS-LLM-Pos*).

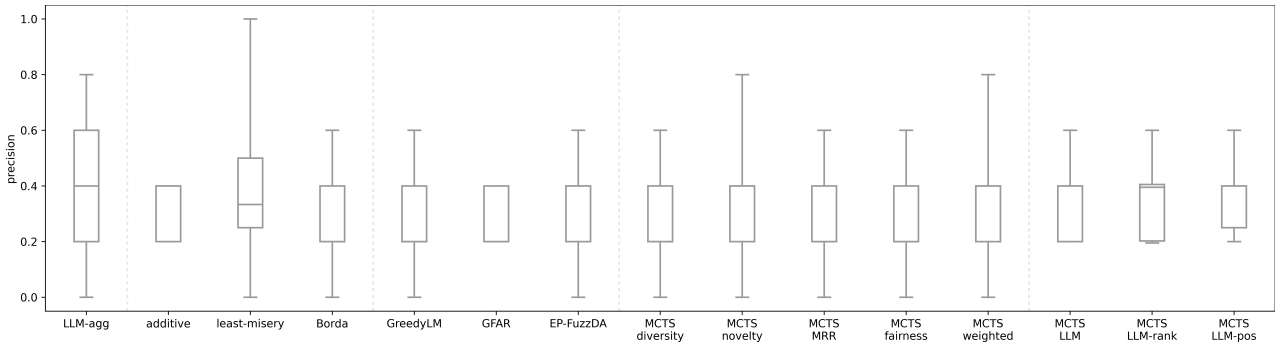
ETHICAL STATEMENT

While LLMs can enhance recommendation accuracy and user experience, they also raise concerns about fairness and bias. Ensuring transparency, accountability, and fairness in LLM-based recommendation systems is crucial for addressing ethical aspects and maintaining user trust.

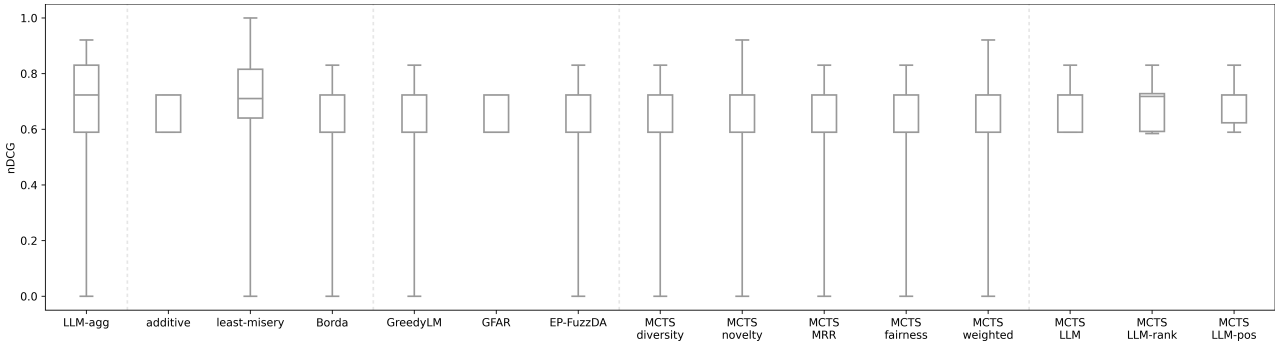
REFERENCES

[1] Linas Baltrunas, Tadas Makcinskas, and Francesco Ricci. 2010. Group recommendations with rank aggregation and collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*. 119–126.

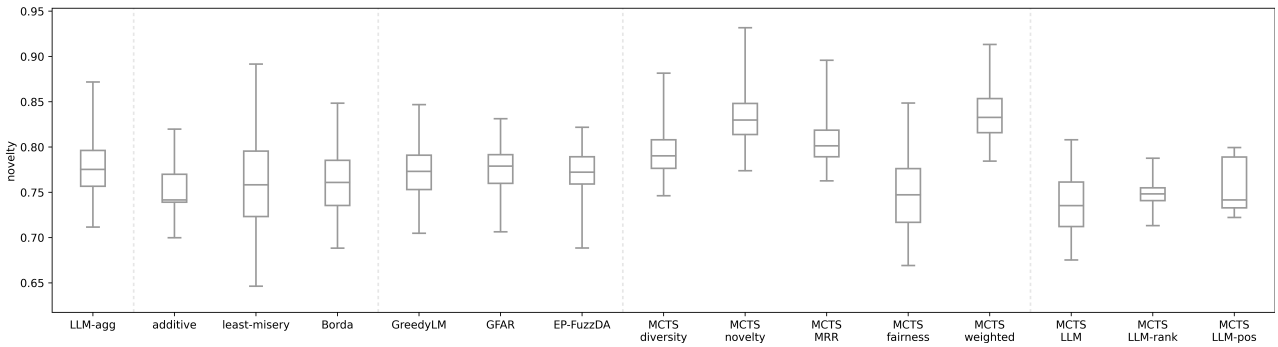
[2] Mesut Kaya, Derek Bridge, and Nava Tintarev. 2020. Ensuring fairness in group recommendations by rank-sensitive balancing of relevance. In *Proceedings of the 14th ACM Conference on recommender systems*. 101–110.



(a) Max precision distribution across groups



(b) Max nDCG distribution across groups



(c) Max novelty distribution across groups

Figure 2: Max metric distribution across groups (similar groups of 5 members)

[3] Ladislav Malecek and Ladislav Peska. 2021. Fairness-preserving group recommendations with user weighting. In *Adjunct Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization*. 4–9.

[4] Judith Masthoff. 2010. Group recommender systems: Combining individual models. In *Recommender systems handbook*. Springer, 677–702.

[5] Allen Nie, Ching-An Cheng, Andrey Kolobov, and Adith Swaminathan. 2023. Importance of Directional Feedback for LLM-based Optimizers. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*. https://www.microsoft.com/en-us/research/publication/importance-of-directional-feedback-for-llm-based-optimizers/

[6] Ladislav Peska and Ladislav Malecek. 2021. Coupled or Decoupled Evaluation for Group Recommendation Methods?. In *Perspectives@ RecSys*.

[7] Dimitris Sacharidis. 2019. Top-n group recommendations with fairness. In *Proceedings of the 34th ACM/SIGAPP symposium on applied computing*. 1663–1670.

[8] Javier Sanz-Cruzado and Pablo Castells. 2018. Enhancing Structural Diversity in Social Networks by Recommending Weak Ties. In *Proceedings of the 12th ACM Conference on Recommender Systems* (Vancouver, British Columbia, Canada) *(RecSys '18)*. ACM, New York, NY, USA, 233–241. https://doi.org/10.1145/3240323.3240371

[9] Javier Sanz-Cruzado and Pablo Castells. 2018. Enhancing structural diversity in social networks by recommending weak ties. In *Proceedings of the 12th ACM conference on recommender systems*. 233–241.

[10] Christoph Trattner, Alan Said, Ludovico Boratto, and Alexander Felfernig. 2023. Evaluating group recommender systems. In *Group recommender systems: an introduction*. Springer, 63–75.

[11] Robbie CM van Aert. 2023. Meta-analyzing partial correlation coefficients using Fisher's z transformation. *Research Synthesis Methods* 14, 5 (2023), 768–773.

[12] David A Walker. 2003. JMASM9: converting Kendall's tau for correlational or meta-analytic analyses. *Journal of Modern Applied Statistical Methods* 2 (2003), 525–530.

[13] Wen Wang, Wei Zhang, Jun Rao, Zhijie Qiu, Bo Zhang, Leyu Lin, and Hongyuan Zha. 2020. Group-aware long-and short-term graph representation learning for sequential group recommendation. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*. 1449–1458.

prec/nDCG/nov	LLM agg	additive	least-misery	Borda	GreedyLM	GFAR	EP-FuzzDA	MCTS diversity	MCTS novelty	MCTS MRR	MCTS fairness	MCTS weighted	MCTS LLM	MCTS LLM-rank	MCTS LLM-pos
LLM-agg	-	M/S/L	-/-/L	-/-/M	M/-/-	-/-/-	-/-/-	M/-/-	-/-/-	M/-/-	M/S/L	-/-/-	-/-/L	-/-/-	-/-/-
additive	-/-/-	-	-/-/-	-/-/-	-/-/-	-/-/-	-/-/-	-/-/-	-/-/-	-/-/-	-/-/-	-/-/-	-/-/-	-/-/-	-/-/-
least-misery	-/-/-	M/S/-	-	S/-/-	M/-/-	S/-/-	-/-/-	M/-/-	-/-/-	S/-/-	M/-/-	-/-/-	-/-/-	-/-/-	-/-/-
Borda	-/-/-	M/M/M	-/S/M	-	M/M/-	S/-/-	-/-/-	S/S/-	-/-/-	S/S/-	M/S/S	-/-/-	-/-/L	-/-/-	-/-/-
GreedyLM	-/-/-	-/-/L	-/-/L	-/-/M	-	-/-/-	-/-/-	-/-/-	-/-/-	-/-/-	-/-/L	-/-/-	-/-/L	-/-/-	-/-/-
GFAR	-/-/-	-/M/L	-/S/L	-/-/L	S/S/-	-	-/-/S	-/S/-	-/-/-	-/-/-	S/M/L	-/-/-	-/-/L	-/-/L	-/-/L
EP-FuzzDA	-/-/-	S/M/L	-/S/L	-/-/M	S/S/-	-/-/-	-	S/S/-	-/-/-	-/-/-	M/M/L	-/-/-	-/-/L	-/-/-	-/-/-
MCTS-diversity	-/-/L	-/-/L	-/-/L	-/-/L	-/-/L	-/-/L	-/-/L	-	-/-/-	-/-/-	-/-/L	-/-/-	-/-/L	-/-/L	-/-/L
MCTS-novelty	-/-/L	M/M/L	-/-/L	-/-/L	M/S/L	-/-/L	-/-/L	M/S/L	-	S/S/L	M/S/L	-/-/-	-/-/L	-/-/L	-/-/L
MCTS-MRR	-/-/L	-/S/L	-/-/L	-/-/L	-/-/L	-/-/L	-/-/L	-/-/M	-/-/-	-	-/-/L	-/-/-	-/-/L	-/-/L	-/-/L
MCTS-fairness	-/-/-	-/-/-	-/-/-	-/-/-	-/-/-	-/-/-	-/-/-	-/-/-	-/-/-	-/-/-	-	-/-/-	-/-/S	-/-/-	-/-/-
MCTS-weighted	-/-/L	M/M/L	-/S/L	-/-/L	M/S/L	-/-/L	-/-/L	S/S/L	-/-/S	S/S/L	S/S/L	-	-/-/L	-/-/L	-/-/L
MCTS-LLM	S/M/-	L/L/-	S/M/-	S/-/-	M/M/-	M/-/-	S/-/-	L/M/-	M/-/-	L/M/-	L/L/-	-/-/-	-	-/-/-	-/-/-
MCTS-LLM-rank	-/-/-	-/-/-	-/-/-	-/-/-	-/-/-	-/-/-	-/-/-	-/-/-	-/-/-	-/-/-	-/-/-	-/-/-	-/-/-	-	-/-/-
MCTS-LLM-pos	L/L/-	L/L/-	L/L/-	-/-/-	-/-/-	-/-/-	-/-/-	-/-/-	L/L/-	-/L/-	L/L/-	-/-/-	-/-/L	-/-/-	-

Table 2: Statistical significance of differences for similar groups of size 5. A cell indicates whether the row was significantly higher (one-tail test) than the column. '-' indicates that differences were not statistically significant. S(mall), M(edium) and L(arge) indicate the corresponding effect size, with $\alpha = 0.05$.

[14] Lin Xiao, Zhang Min, Zhang Yongfeng, Gu Zhaoquan, Liu Yiqun, and Ma Shaoping. 2017. Fairness-aware group recommendation with pareto-efficiency. In *Proceedings of the eleventh ACM conference on recommender systems*. 107–115.

[15] Zirui Zhao, Wee Sun Lee, and David Hsu. 2024. Large language models as commonsense knowledge for large-scale task planning. *Advances in Neural Information Processing Systems* 36 (2024).