WINNING WITHOUT COMPOSING, LOSING WHILE GENERALIZING: SEPARATING BETTER SCORES FROM BETTER CHOICES IN RECOMMENDATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recommendation models may need to suggest items from unseen combinations of familiar attributes, such as brand and product category. We examine whether withholding interactions with selected attribute combinations, while retaining their constituent attributes elsewhere in training, provides evidence of compositional generalization. A score that credits only the withheld group can reward recommending more items from that group rather than choosing better items within it. We introduce a model-agnostic audit that keeps the number and positions of group recommendations fixed for each user while changing which items fill those positions. When both models rank every item, our audit splits the difference in their recall exactly into a part due to how many group items each recommends and a part due to which group items it chooses. In experiments with synthetic data and known user preferences, a preference-blind rule that recommends more items from the withheld combination can win the group score against models that rank its unseen items better. Real-data comparisons reveal both cases where higher performance is driven mainly by recommending more items from the target group and cases where the advantage persists after matching the number and positions of recommendations from that group. We also use the audit to select group-specific ranking rules using one set of users, then test the fixed rules on additional users without changing group positions. The correction improves mean accuracy across the complete recommendation lists, beyond both the original model and a single combination applied to all groups. Consequently, our audit distinguishes whether a model recommends more items from a group or chooses better items within it, providing a more informative basis for evaluating compositional generalization and testing targeted improvements.

1 INTRODUCTION

Suppose a recommender learns about a brand and a category from different products. Can it correctly recommend products from their unseen combination? This is a form of compositional generalization, where the constituents of a combination (e.g., Sony and headphones) are observed during training, while interactions with the combination (e.g., Sony headphones) are withheld. Learning representations that capture reusable factors can enable models to generalize to novel combinations of familiar components (Bengio et al., 2013; LeCun et al., 2015). Disentangled collaborative filtering aims to learn such factors from interaction data (Ma et al., 2019; Wang et al., 2020; Ren et al., 2023).

When a model recommends from the full catalogue but receives credit only for relevant items from the withheld group, its score combines two decisions: how many positions it gives the group (*allocation*), and which group items it chooses (*ordering*). A model can therefore achieve a higher group-level score simply by allocating more positions to the target group, even if it makes worse choices among the items in that group.

Figure 1 contrasts the two setups. The catalogue lists give Model A more group positions than Model B. Filtering each model's full ranking to Sony headphones and taking its first five gives both

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

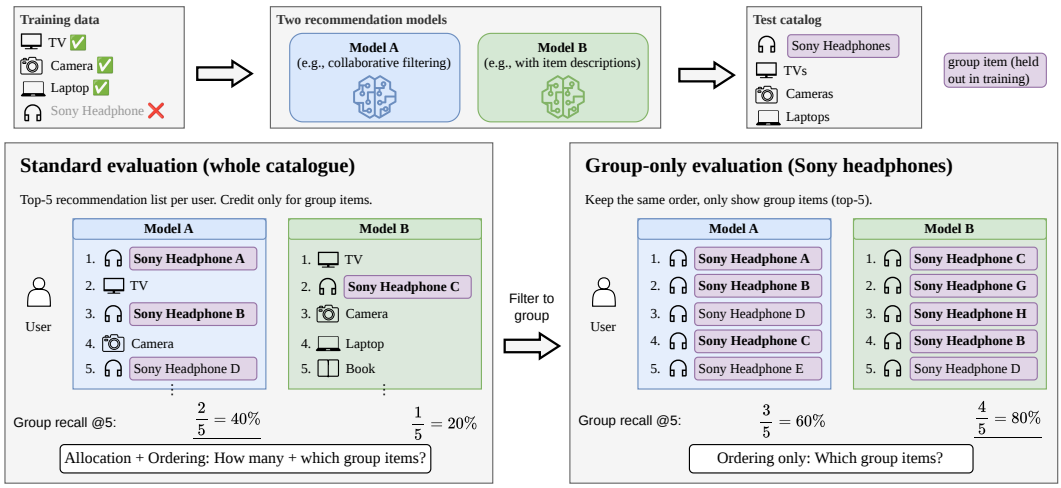

Figure 1: Catalogue and group-only evaluations ask different questions. The five relevant headphones are shown in bold. The catalogue evaluation scores the model's original top-five list. The group-only comparison takes the first five Sony headphones from its full ranking. Group recall counts retrieved relevant headphones, not every headphone on the list. The same relevant set must be used in both comparisons.

the same group count. We can then ask which model retrieves more relevant headphones when both receive five opportunities to do so.

Recommendation scores can conceal allocation and ranking differences (Cañamares & Castells, 2018; Steck, 2018; Li et al., 2023b). Ding et al. (2026) report scores on generalization cases defined from user histories. Such scores still combine group placement with item choice.

Our model-agnostic *audit* asks whether a score advantage survives when both models receive the same opportunities to recommend from the group. For each user, it keeps the group's recommendation positions fixed and exchanges the items that fill them, preserving each model's within-group order. With complete rankings, it splits the recall gap exactly into allocation and ordering components. When generated lists reveal only part of a ranking, it bounds comparisons that require the missing choices. Separate symmetry results establish when random ordering is a justified reference for items with no training interactions.

Experiments with synthetic data show why the distinction matters: a preference-blind rule can win group recall while models that usefully rank unseen items lose. Equal group counts reveal their better item choices. On recorded data, extra group recommendations explain most of one model's advantage, whereas the published generalization advantage survives matched positions on the Beauty and Steam datasets. The audit can therefore challenge a score advantage or support it.

The audit also guides a practical correction. We select group-specific combinations of existing rankings using one set of users, freeze the rules, and test them on additional users in the Steam dataset. At unchanged group positions, the correction improves mean complete-list accuracy beyond both the original recommender and a combination selected on the same users and shared by all groups. Together, these results give a procedure for interpreting group scores and testing targeted improvements.

2 Recommendation Task and Generalization

We define our recommendation setting and the relevant concepts for compositional generalization.

Recommendation. Let $\mathcal{U}$ and $\mathcal{I}$ be finite sets of users and items and let D contain the recorded interactions (u, i) (e.g., purchases). $P \subseteq \mathcal{I}$ is the candidate pool of items considered for recommendation and $E_u \subseteq P$ are those candidate items eligible to the user u according to the evaluation rules, for example after removing items already in the user's training history. A recommendation model

yields user-item scores as $\mathcal{U} \times \mathcal{I} \to \mathbb{R}$. Given a user u and a cut-off $k \geq 1$, the model A creates an ordered recommendation list $L_{A,u,k}$ which consists of the $\min(k, |E_u|)$ eligible items with highest score, with ties resolved by a declared rule.

Evaluation. We define $R_u \subseteq \mathcal{I}$ as the *answer key* that consists of items relevant to user u during testing. Given a recommendation list $L_{A,u,k}$, Recall@$k = \frac{|R_u \cap L_{A,u,k}|}{|R_u|}$ measures the fraction of relevant items retrieved, for $|R_u| > 0$. NDCG@k gives more credit for retrieving a relevant item earlier in the list normalized by the ideal ordering (Järvelin & Kekäläinen, 2002).

Attribute groups and group recall. An attribute combination specifies values such as brand = Sony and category = headphones. The target group $C \subseteq \mathcal{I}$ contains the items with that combination. Group recall applies the same formula with $R_u \cap C$ replacing R_u, provided this intersection is nonempty. The list may still include items outside C.

The eligible group for user u is $C_u = C \cap E_u$. More generally, C_u may depend on the user's history, but not on the test answer.

Attributes, factors and composition. We distinguish between observed item attributes and latent factors learned from interaction data.

Definition 1 (Model- and user-novel combinations)**.** An attribute or learned-factor combination is *model-novel* if it does not occur in the training interactions, whereas it is only *user-novel* if it is absent for a particular user but observed for others.

Definition 2 (Cold and warm items)**.** An item is *training-cold* if not part of any training interaction, and *warm* otherwise. A group is fully training-cold when all its items are training-cold.

Definition 3 (Learned-factor composition & Feature composition)**.** *Learned-factor composition* ranks highly an item combining latent, interaction-discovered factors in a model-novel way, after learning each factor in other combinations. *Feature composition* instead scores an unseen combination of observed attributes by composing their trained embeddings.

3 Allocation, item choice and group scores

Suppose one model has higher group recall than another. We want to know whether it chooses better items or simply recommends more items from the target group.

3.1 Two decisions hidden in one score

We distinguish how many group items enter a recommendation list (*allocation*) from the model's relative ranking of group items (*ordering*): two separate decisions typically hidden in a single scoring function.

Recall within a group. To compare item choices, we score the same number of group recommendations per user, preserving each model's within-group order. For user u, $r_u(l)$ is ordinary recall from the first l items in this restricted ranking. Here l counts group items, whereas k counts all positions in the original recommendation list. Filtering a ranking to the group preserves its prefixes (Zehlike et al., 2017, Lemma 4.3), giving an exact recall identity.

Proposition 1 (within-group prefix identity)**.** *Let E_u be a user's eligible items, $C_u \subseteq E_u$ a per-user group, and $\varnothing \neq R_u \subseteq C_u$. Fix one strict total order on E_u, the realized ranking after applying the declared tie rule. Let L_u count group items in its first $\min(k, |E_u|)$ positions and let $r_u(l)$ be recall in the first l positions of the* same *order restricted to C_u, using denominator $|R_u|$. Then*

$$\text{Recall}_u@k = r_u(L_u).$$

Proposition 1 is for recall. NDCG also rewards earlier hits, so we hold positions fixed.

Accounting for a score difference. Suppose model A gives a user two group slots and model B gives five. Scoring each model's first two and first five group choices gives four comparisons.

Let $r_{A,u}(l)$ be recall from model A's first l group choices for user u, and let $L_{B,u} = |L_{B,u,k} \cap C_u|$ count group items in B's top-k list. In this example, $r_{A,u}(L_{B,u})$ scores A's first five group choices. We borrow B's count, not its chosen items. Averaging these userwise recalls gives q_{AB}:

$$q_{AB} = \text{mean}_u \, r_{A,u}(L_{B,u}). \tag{1}$$

Crossing the two models' item orders and group counts gives four means: $q_{AA}, q_{AB}, q_{BA}, q_{BB}$ (Figure 3). All use the same users with at least one relevant group item. For each user, both within-group orders must reach the requested counts $L_{A,u}$ and $L_{B,u}$. Complete rankings of the common eligible set suffice. Insufficient prefixes instead require completion bounds (Section 6.3). The original recalls are q_{AA} and q_{BB}. For allocation, we change group counts at a fixed within-group ranking and average over both rankings. For ordering, we change rankings at fixed counts and average over both models' counts. Neither model is the sole reference:

$$\mathcal{A} = \tfrac{1}{2}\big[(q_{AA} - q_{AB}) + (q_{BA} - q_{BB})\big], \qquad \text{allocation component,} \tag{2}$$

$$\mathcal{D} = \tfrac{1}{2}\big[(q_{AA} - q_{BA}) + (q_{AB} - q_{BB})\big], \qquad \text{ordering component.} \tag{3}$$

The original gap is exactly $q_{AA} - q_{BB} = \mathcal{A} + \mathcal{D}$. Positive components favour A in this mean-score comparison. The decomposition concerns recommendation outcomes, not model internals. We calculate it user by user before averaging because group counts vary across users. Mean recall curves alone cannot recover the components or show which model is better for every user.

3.2 Chance controls and candidate pools

A uniformly random order of $N > 0$ eligible group items containing a fixed nonempty relevant set has expected recall ℓ/N from its first ℓ items (Proposition 6). This provides a preference-blind reference at matched group counts. Applying it as a learned-model null requires ranked-output symmetry, including selection and ties (Appendix A.7). The audit itself needs no symmetry assumption.

Changing candidates while keeping rankings and relevant items fixed tests dependence on the pool. Appendix B.3 gives the formal implications of reversals. Moving a group up also preserves its internal order while potentially displacing useful items (Remark 1). We therefore grade complete lists when testing a correction.

4 Experimental design

Does a group-score winner choose better items when its opportunities to recommend from the group are controlled? We use known preferences in synthetic data and recorded interactions in logged data. Table 3 in Appendix A gives constructions and fitting budgets.

4.1 Synthetic worlds with a utility reference

Synthetic data creation. We generate $1,000$ users and $1,000$ items with three latent factors to evaluate recommendations against known preferences. User-item utilities are computed by matching user and item factor vectors (details in Appendix A.2). We then add noise to the utilities and record the twenty highest-scoring items per user as observed interactions. The *oracle* ranks items by their noise-free utilities. The trained models do not observe the latent factors or utility values. Shared reference vectors, called *prototypes*, induce discrete feature tags. Tags from two factors define the withheld combination and its item group. The third factor affects preferences but not group membership.

Models and training. In our simulation, we consider three recommendation methods. Interaction predicts from the observed user–item links. Its item scores do not use the prototype tags. For each factor, the generator also links every user and item to its five nearest prototypes. Separate receives these prototype identifiers as feature tags and processes each factor's links in a separate channel. An item without training interactions still has these tags. Mixing lets information from the channels mix through learned gates. Both learn representations of the supplied IDs. Architecture, fitting and code IDs are in Appendix A.2. We compare the original catalogue lists with each model's highest-ranked group items, using the oracle's ten highest-ranked eligible items as the preference reference.

Holdouts and repetitions. The main experiment removes all group interactions from fitting and validation targets. Each model has five split/training runs within the same generated world. Appendix A.2 also reports the original comparison across cold and warm constructions.

4.2 Logged Data & Comparison Models

Datasets. We fit models on Amazon Electronics and HetRec, then audit released models on Beauty and Steam. For Amazon Electronics and HetRec, each retained rating counts as an interaction regardless of its value.

Amazon: The Amazon Reviews 2023 Electronics dataset contains product-review records with brand and category information (Hou et al., 2026). We withhold selected users' interactions with three brand–category groups, referred to by brand as Logitech, Sony and Canon. Other users retain training interactions with these combinations, so they are not model-novel.

HetRec: The HetRec2011 dataset contains movie ratings enriched with genre and country information (Cantador et al., 2011). Our target group consists of films labelled both Comedy and UK. Removing selected users' group interactions leaves some films warm and the others training-cold (Appendix A.4).

Beauty and Steam contain beauty-product and video-game interaction histories, respectively. We use the released next-item task of Ding et al. (2026). Its groups depend on each user's history rather than withheld attribute combinations.

Models for Amazon and HetRec. We use the following models on Amazon (BPR, LightGCN and SB-Pop) and HetRec (BPR, LightGCN, SB-Pop, ItemKNN, DGCF and MacridVAE). These methods receive interactions, not the attributes used to define the evaluation groups.

- **BPR** learns user and item vectors to rank recorded interactions above sampled unobserved items (Rendle et al., 2009).
- **LightGCN** learns user and item representations by passing information along the observed links in a user–item graph (He et al., 2020).
- **SB-Pop** is RecBole's non-personalized `Pop` rule, whose item counter includes both recorded interactions and sampled unobserved items in training batches (Zhao et al., 2021).
- **ItemKNN** ranks items by their similarity to a user's history items, with similarity derived from users' shared interactions.
- **DGCF** is a graph recommender that learns separate components intended to represent different user interests (Wang et al., 2020).
- **MacridVAE** uses a variational autoencoder to reconstruct users' interaction histories through separate preference components (Ma et al., 2019).

Models for Beauty and Steam. SASRec ranks catalogue items from a user's interaction history (Kang & McAuley, 2018). TIGER generates identifiers for recommended items and returns a short list (Rajput et al., 2023). We replay the checkpoints released with that study rather than refitting them. An untrained item-similarity rule provides an additional control in Section 6.3.

5 Results on Synthetic Data

Can a rule that does not learn preferences beat models with useful rankings of training-cold items? Figure 2 compares their original lists and lists with equal group counts in the fully cold construction, using the known-preference reference.

The blind rule specializes in the evaluation target. The withheld group is exactly the set of items with no training interactions. A rule can therefore identify it without learning anyone's preferences. Putting those items first in a random order shared across users gives all ten positions to the group, maximizing the uniform-order expected recall (Proposition 6). Separate and Mixing average fewer than one group item per top-ten list. The blind rule's expectation and realization beat both models' original group scores (panel A).

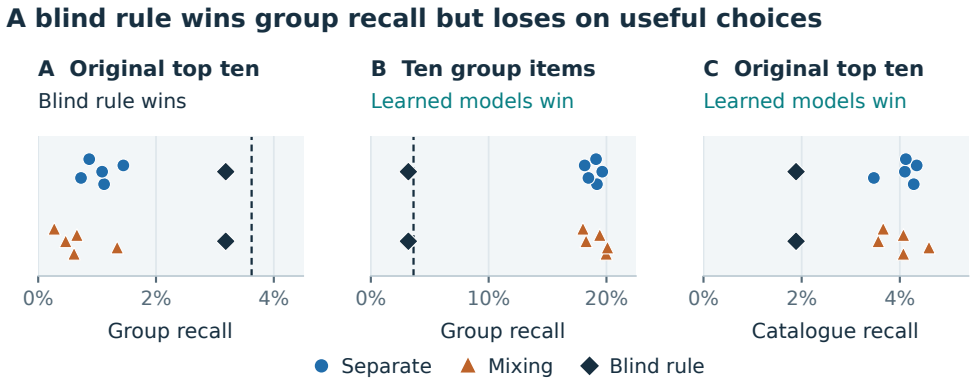

Figure 2: **A blind rule wins group recall but loses on item choice and catalogue recall.** Circles and triangles show five training runs per model, each averaging Recall@10 over the same 997 users with a recorded withheld-group interaction in one additive-preference world. **A** scores the original top-ten lists on preferred group items. **B** scores each model's first ten group choices without retraining. **C** scores the original lists on all preferred eligible items. Each panel repeats one fixed blind-rule score as a diamond on both model rows. Dashed lines in A and B show its expectation. Vertical offsets only separate marks. Horizontal scales differ. Appendix A.2 shows both preference rules.

Equal slots reveal useful ordering. Taking each saved ranking's first ten group choices, every feature-aware checkpoint scores several times the random expectation at that *same count* (panel B). INTERACTION is not plotted. At ten group slots, INTERACTION's tied cold-item scores give the blind rule's shared random order. Their original policies differ: the blind rule explicitly puts cold items first. The native group-score loss concealed useful unseen-item ranking. Both feature-aware models beat the realized blind rule on catalogue recall (panel C).

Sensitivity to the synthetic construction. Adding training-cold items outside the group removes the shortcut of identifying group membership from zero training degree. Useful discrimination of relevant items persists.

All three comparisons also hold with cross-term preferences and on two additional draws from the same generator (Appendix A.2).

The original stress test. Group-only NDCG@10 against recorded interactions never put first the trained model ranked last by catalogue Recall@10 against preferred group items. This held in thirty original cold/warm comparisons under both preference rules (Appendix A.2). Together, these synthetic results show that a group-score loss can conceal useful ranking of unseen items, and that matching group counts reveals it.

6 RESULTS ON LOGGED DATA

On logged data, we examine three cases: an advantage mainly due to more group recommendations, no recommendations of training-cold items, and an advantage that survives matched positions.

6.1 PRODUCT RECOMMENDATION: CANDIDATE POOLS AND GROUP POSITIONS

Why does BPR beat LightGCN on catalogue Recall@20 for the Logitech group in Amazon Electronics? Crossing their item choices and per-user group counts gives the four scores in Figure 3. Most of BPR's advantage comes from recommending more group items. The ordering component's interval includes zero.

Changing the competing items. With the same fitted models and relevant items, we compare the group and catalogue pools with two enlarged pools. *Confusers* share the target brand or category.

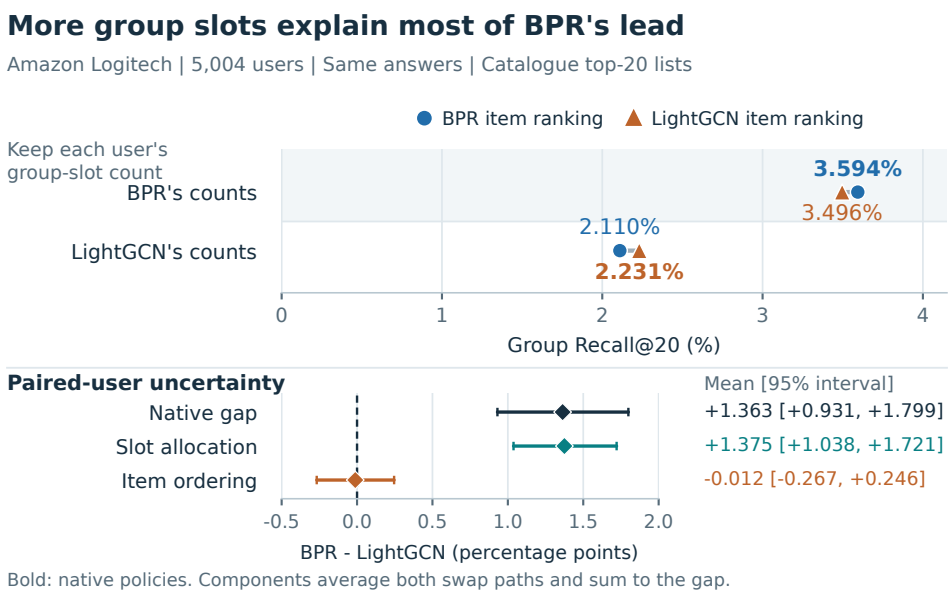

Figure 3: **More group slots explain most of BPR's advantage.** Rows fix each user's group-slot count to BPR or LightGCN, then compare their frozen within-group orders. Points are mean group Recall@20. Bold values mark native policies. Below, components average both swap paths and sum to the native gap. Bars are pointwise 95% paired-user intervals, conditional on the saved checkpoints and split. The ordering interval includes zero. Methods: Appendix A.3.

Random pools match the confuser pools' size before excluding each user's history. On Logitech Recall@20, the group-only pool favours LightGCN, both enlarged matched-size pools favour user-blind SB-POP, and the catalogue favours BPR (Table 5 in Appendix A.3).

We approximately match random candidates to confusers by training interaction count. This closes about four fifths of BPR and LightGCN's NDCG@10 gap between these pools (Table 5).

Appendix A.3 gives the redraws, fixed-order reversal checks and the Sony and Canon results, including Canon's absent Recall@20 reversal.

6.2 MOVIE RECOMMENDATION: THE COLD ITEMS NEVER REACHED

The HetRec study asks what group scores reveal about cold-item choices. None of the six original fitted methods recommends a training-cold film in its catalogue top twenty. With no cold recommendations, the score cannot reveal which cold items the model would select if given slots. Even on the group-only pool, LightGCN's and ItemKNN's hits are all films that retain training interactions. Appendix A.4 gives the census of cold and residually warm items.

6.3 AUDITING A PUBLISHED GENERALIZATION ADVANTAGE

Question and setting. Can the audit also support a reported advantage in item choice?

The generalization group of Ding et al. (2026) contains items reachable through a training-derived history graph, excluding recorded one-step successors of the last item. Membership uses no test target.

Comparison. Each cohort retains every user whose next recorded item belongs to the generalization group, including misses and zero group counts. Previously seen items remain eligible as in the released task. To compare choices, fill TIGER's valid, distinct group positions with SASRec's group choices. Positions 2 and 7, for example, stay 2 and 7 rather than becoming 1 and 2. Other positions remain uncredited. This comparison exchanges items in saved lists, not tokens inside the decoder.

Results. Table 1 separates the two gains: better placement, then better choices at those positions. Both have positive 95% paired-user intervals in each domain, conditional on these checkpoints, items and split (Appendix A.5).

Table 1: **TIGER's advantage survives matched group positions.** Mean NDCG@10 on generalization-target users. The last two rows keep each user's group positions fixed.

Item choices	Group positions	Beauty ($n = 18{,}170$)	Steam ($n = 29{,}354$)
SASRec	SASRec	0.01343	0.01330
SASRec	TIGER	0.01609	0.01379
TIGER	TIGER	0.02095	0.01568

TIGER's advantage includes better item choices, not just more or earlier group recommendations. Reversing the swap puts TIGER's choices at SASRec's group positions, but TIGER's short list may reveal too few items to fill them. Even the least favourable completion consistent with TIGER's observed group prefix retains a positive mean advantage (Appendix A.5). Better item choices here do not distinguish composition from memorization.

Item-similarity control. As a cheap control on the same cohorts, we fill TIGER's group positions using summed training-derived cosine similarities to distinct history items, prioritizing items absent from the history. TIGER retains a positive paired mean advantage on both domains. SASRec also wins on Steam but averages below the control on Beauty, with a difference interval containing zero. Appendix A.5 gives the full control.

7 Using the audit to guide a correction

We test whether combining existing rankings differently for each history-defined group improves complete lists beyond one selected combination used everywhere.

Selecting and testing the correction. On previously examined Steam users, we selected combinations of TIGER, SASRec and item-similarity rankings separately for three groups: recorded one-step successors of the user's last item, the study's history-reachable group, and remaining items. Groups use the training graph and the user's history, not test answers. We combine reciprocal ranks using established rank fusion (Cormack et al., 2009).

Selection maximizes full-list NDCG@10 at TIGER's fixed group positions on the same users for both rules. The global control selects one combination for all groups from the same ranking sources, also considering rules that favour or demote previously seen items (Appendix A.6). Selection returns TIGER plus half-weight SASRec as the global rule. The correction leaves it unchanged for one-step successors, adds half-weight item similarity for history-reachable items and uses SASRec plus item similarity elsewhere.

We froze rules and tests before evaluating 46,790 additional history-eligible Steam users from the same source and period, not used to select or assess the correction. Full-list NDCG@10 credits the next recorded item in any group. Every list retains the same per-user group positions. Differences therefore reflect item choices, not extra or earlier group recommendations. Fixing group positions does not fix where previously seen items appear.

Table 2: **The correction improves full-list NDCG@10 at unchanged group positions.** All rows use the same 46,790 additional Steam users. Differences use unrounded means. Intervals have nominal 95% coverage across the two predeclared primary contrasts after Bonferroni adjustment, conditional on the saved models and cohort.

List at native TIGER group positions	NDCG@10	Correction − row
Native TIGER	.154304	$+.002233$ $[+.001458, +.003009]$
Selected global combination	.155220	$+.001318$ $[+.000870, +.001767]$
Frozen group-specific correction	.156538	

Results and interpretation. The correction improves mean full-list NDCG@10 by a relative 1.45% over native TIGER and 0.85% over the selected global control (Table 2). All of the measured gain over the global control arises for answers in the audited history-reachable group (Appendix A.6).

Subgroups and changed placements. Users whose next item repeats their history lose on average against both comparators. Gains on next items absent from the history outweigh these losses in the mean. Without retuning, the correction also beats the global control under three other specified placements, using the same cohort and trained models (Appendix A.6). On Beauty, the same selection procedure did not establish an additional benefit over its global control.

Applying the audit. To explain why one recommender scores higher, the audit compares item choices at matched positions in groups defined without test answers. A surviving advantage supports better choices there. If it disappears, placement can explain it. When native lists contain no group items, deeper rankings can supply choices at a common count. Missing choices require completion bounds or an unresolved comparison. Proposed corrections are selected on one set of users, frozen, then tested on separate users using complete-list scores. Appendix A.7 adds chance and candidate-pool checks.

8 Related Work

Offline metrics can misrank recommenders (Cañamares & Castells, 2018), candidate sampling can reverse comparisons (Krichene & Rendle, 2020), and some sequential benchmarks admit shortcuts (Han et al., 2026).

Repeat/explore studies compare allocation with within-task accuracy (Li et al., 2023a). Removing repeats without refitting can reverse pure-exploration comparisons (Li et al., 2023b). TREx separates repetition and exploration components (Li et al., 2024), while repeat-bias-aware optimization sets per-user group budgets before item selection (Liu et al., 2025). Calibration, component-wise comparisons and behavioral tests provide precedents (Steck, 2018; Jeon et al., 2024; Beutel et al., 2019; Wang et al., 2021; Chia et al., 2022).

MacridVAE, DGCF and DCCF assess recommendation accuracy using Recall and NDCG (Ma et al., 2019; Wang et al., 2020; Ren et al., 2023). Our audit compares saved item choices at matched group counts for recall and positions for NDCG. Completion bounds handle missing ranks, while symmetry analysis supplies cold-item chance and relabeling benchmarks (Appendix A.7).

Content-enabled cold start (Volkovs et al., 2017) and held-out combinations (Lake & Baroni, 2018; Hupkes et al., 2020) are established. Disentanglement need not predict composition (Xu et al., 2022). Our conditional statistical guarantee uses classical finite-group and add-one tests (Hemerik & Goeman, 2018; Phipson & Smyth, 2010).

9 Implications and limitations

Our audit reveals useful ranking concealed by lower group scores, identifies advantages driven mainly by extra group recommendations, and supports advantages that survive matched group positions. A correction selected on one set of users improved mean complete-list accuracy for additional users in the Steam dataset beyond both the original model and a selected global combination, without changing group positions.

These comparisons do not isolate the effects of model architecture. Evidence covers one generator family with jointly developed models, selected logged cases and two domains of one published study.

Benchmarking tools could report matched-position audits and frozen-correction tests alongside native scores. The audit connects score differences to explanations and targeted improvements.

REPRODUCIBILITY STATEMENT

Appendix A gives the experimental protocols, and Appendix B gives the assumptions and proofs. The anonymous repository[1] provides synthetic-data generators, model-fitting code for our synthetic and logged-data experiments, and code for auditing released checkpoints and evaluating the correction. Its study guides give environment settings, commands, and links to public datasets and released checkpoints. Fresh fitting and analysis of released checkpoints are documented separately.

REFERENCES

Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8), 2013.

Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Li Wei, Yi Wu, Lukasz Heldt, Zhe Zhao, Lichan Hong, Ed H. Chi, and Cristos Goodrow. Fairness in recommendation ranking through pairwise comparisons. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2019.

Rocío Cañamares and Pablo Castells. Should I follow the crowd? A probabilistic analysis of the effectiveness of popularity in recommender systems. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2018.

Iván Cantador, Peter Brusilovsky, and Tsvi Kuflik. Second workshop on information heterogeneity and fusion in recommender systems (HetRec2011). In *Proceedings of the ACM Conference on Recommender Systems*, 2011.

Patrick John Chia, Jacopo Tagliabue, Federico Bianchi, Chloe He, and Brian Ko. Beyond NDCG: Behavioral testing of recommender systems with RecList. In *Companion Proceedings of the Web Conference*, 2022.

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Büttcher. Reciprocal rank fusion outperforms Condorcet and individual rank learning methods. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2009.

Yijie Ding, Zitian Guo, Jiacheng Li, Letian Peng, Shuai Shao, Wei Shao, Xiaoqiang Luo, Luke Simon, Jingbo Shang, Julian McAuley, and Yupeng Hou. How well does generative recommendation generalize? arXiv preprint arXiv:2603.19809v1, 2026. The source version used for the reported replay. Published as *On the Memorization and Generalization of Generative Recommendation*, Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (2026).

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, 2017.

Haoyu Han, Li Ma, Hanbing Wang, Bingheng Li, Daochen Zha, Chun How Tan, Huiji Gao, Xin Liu, Stephanie Moyerman, Sanjeev Katariya, Hui Liu, and Jiliang Tang. An embarrassingly simple graph heuristic reveals shortcut-solvable benchmarks for sequential recommendation. arXiv preprint arXiv:2605.07125, 2026.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. LightGCN: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020.

Jesse Hemerik and Jelle Goeman. Exact testing with random permutations. *TEST*, 27(4), 2018.

Yupeng Hou, Jiacheng Li, Xiangjun Fu, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. Bridging language and items for retrieval and recommendation: Benchmarking LLMs as semantic encoders. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2026. Amazon Reviews 2023 dataset; cited source version: arXiv:2403.03952v2 (20 April 2026).

[1] `https://anonymous.4open.science/r/recommendation-audit-review-26F1/README.md`

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research*, 67, 2020.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems*, 20(4), 2002.

Hyunsik Jeon, Se-eun Yoon, and Julian McAuley. Calibration-disentangled learning and relevance-prioritized reranking for calibrated sequential recommendation. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, 2024.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *Proceedings of the IEEE International Conference on Data Mining*, 2018.

Walid Krichene and Steffen Rendle. On sampled metrics for item recommendation. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2020.

Brenden M. Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of the International Conference on Machine Learning*, 2018.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553), 2015.

Ming Li, Sami Jullien, Mozhdeh Ariannezhad, and Maarten de Rijke. A next basket recommendation reality check. *ACM Transactions on Information Systems*, 41(4), 2023a.

Ming Li, Ali Vardasbi, Andrew Yates, and Maarten de Rijke. Repetition and exploration in sequential recommendation. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023b.

Ming Li, Yuanna Liu, Sami Jullien, Mozhdeh Ariannezhad, Mohammad Aliannejadi, Andrew Yates, and Maarten de Rijke. Are we really achieving better beyond-accuracy performance in next basket recommendation? In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024.

Yuanna Liu, Ming Li, Mohammad Aliannejadi, and Maarten de Rijke. Repeat-bias-aware optimization of beyond-accuracy metrics for next basket recommendation. In *European Conference on Information Retrieval*, 2025.

Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. Learning disentangled representations for recommendation. In *Advances in Neural Information Processing Systems*, 2019.

Alistair Moffat, Joel Mackenzie, Antonio Mallia, and Matthias Petri. A flexible resource for top-weighted comparisons between sets and rankings. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2025.

Belinda Phipson and Gordon K. Smyth. Permutation P-values should never be zero: calculating exact P-values when permutations are randomly drawn. *Statistical Applications in Genetics and Molecular Biology*, 9(1), 2010.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Maheswaran Sathiamoorthy. Recommender systems with generative retrieval. In *Advances in Neural Information Processing Systems*, 2023.

Xubin Ren, Lianghao Xia, Jiashu Zhao, Dawei Yin, and Chao Huang. Disentangled contrastive collaborative filtering. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2009.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, 2018.

Ashudeep Singh and Thorsten Joachims. Fairness of exposure in rankings. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2018.

Harald Steck. Calibrated recommendations. In *Proceedings of the ACM Conference on Recommender Systems*, 2018.

Maksims Volkovs, Guangwei Yu, and Tomi Poutanen. DropoutNet: Addressing cold start in recommender systems. In *Advances in Neural Information Processing Systems*, 2017.

Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. Disentangled graph collaborative filtering. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020.

Xuezhi Wang, Nithum Thain, Anu Sinha, Flavien Prost, Ed H. Chi, Jilin Chen, and Alex Beutel. Practical compositional fairness: Understanding fairness in multi-component recommender systems. In *Proceedings of the ACM International Conference on Web Search and Data Mining*, 2021.

Zhenlin Xu, Marc Niethammer, and Colin Raffel. Compositional generalization in unsupervised compositional representation learning: A study on disentanglement and emergent language. In *Advances in Neural Information Processing Systems*, 2022.

Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. FA*IR: A fair top-k ranking algorithm. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, 2017.

Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. RecBole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, 2021.

AI USE

OpenAI Codex and Anthropic Claude assisted with proposing and refining research ideas and experiments, developing and checking mathematical claims and proofs, implementing synthetic-data generation and data processing, writing and checking model, audit and reproduction code, interpreting results, literature search and source review, figure design, and drafting and editing this manuscript. The authors are responsible for the final text, claims, code and results.

A EXPERIMENTAL DETAILS AND SUPPORTING EVIDENCE

A.1 CONSTRUCTIONS, FITTING AND SCOPE

For Amazon and HetRec, we ignore rating values and treat each retained user–item rating as an observed interaction. Table 3 distinguishes the holdouts and recorded budgets. The models use different inputs and fitting budgets, rather than forming a common-budget architecture comparison. Evaluation masks training and validation history except in the released sequential task, whose original unmasked-history policy is retained. The reproduction package's study guides under `experiments/` map the reported comparisons to configurations, inputs and commands. Saved numerical references are under `results/`. The guides distinguish fresh fitting from analysis of released checkpoints. The latter does not reproduce the checkpoints' original training.

Cold-input and selection conditions. We inspect the conditions needed for the cold-item symmetry results. Inspected ID-only fitting paths have i.i.d. initialization, conditionally exchangeable sampling and equivariant losses and updates in intended arithmetic (Lemma 1), not certified historical bitwise execution. INTERACTION (M0) has directed context edges with no return path to its user/item readout. Removing the six prototype-context edge sets leaves the user and item representations unchanged in the checked checkpoint. RecBole selection monitors validation NDCG@20,

Table 3: **Holdouts and recorded settings.** S1/S2 share one original world. Logged groups are selected cases. Recorded settings are heterogeneous, not recommended defaults. For gradient-trained models, "cap" means maximum training epochs. Detailed serialized fields remain in the extended record.

Construction	Holdout / recorded fitting
Synthetic cold	Entire factor group absent as fitting/validation targets.
Synthetic warm	All qualified users' group edges (warm-degenerate, about 1% retained), or all group edges for a random tenth of qualified users with group interactions (viable, every group item warm).
Amazon	Logitech/Sony/Canon combinations remain represented in training. BPR cap 15. LightGCN cap 15 except Logitech 50. LightGCN regularizer 10^{-5} except Logitech 10^{-4}.
HetRec near-cold	Removes all qualified users' group edges. BPR/DGCF caps 100, LightGCN/MacridVAE 300. The supplementary BPR runs instead use a 300-epoch cap.
Logged shared fields	Standard configurations: batch 4,096, learning rate 0.001. Validation NDCG@20 checked every five epochs. ItemKNN: $k = 100$, shrinkage zero. SB-POP accumulates counts in one sampled-batch pass. Its recorded cap is not a fitting budget.

updates on ties, and uses index-returning top-k. Removing validation-exposed targets does not make this selector invariant: cold candidates and their tie ordering can still affect selection. The historical selected-model symmetry hypothesis remains unverified.

A.2 SYNTHETIC WORLDS

Preference rules and holdouts. Each user's preference for an item is represented by its utility:

$$v_w(u,i) = \sum_{q\in\{a,b,c\}} \langle x_u^q, x_i^q\rangle + \alpha_w \langle x_u^a, x_i^c\rangle, \qquad \alpha_{S1} = 0, \quad \alpha_{S2} = \tfrac{1}{2}. \tag{4}$$

S1 adds the three matching-factor preferences. S2 also makes the user's a factor interact with the item's c factor. Shared reference vectors, called prototypes, provide feature tags for users and items. Fixed partitions of an item's nearest a and c prototypes define its group, while b affects utility but not membership. S1/S2 share factor, prototype and noise draws, context edges and group labels. They are two utility rules, not independent replications.

The cold construction removes the group's interactions from fitting and validation targets. Two warm constructions retain group interactions, one with sparse support and one keeping every group item warm (Appendix A.1). These three constructions under S1 and S2 give six experimental blocks. Five split/training seeds in each block give thirty comparisons of the three models in the original world. Pools contain the group, the group plus items sharing a defining factor (confusers), or the catalogue. These pools vary both the number and kind of competing items.

Generator and models. Equation 4 uses 16-dimensional normalized Gaussian user, item and prototype vectors, with fifty prototypes per factor. Gaussian noise with standard deviation 0.1 precedes top-twenty interaction selection. Each user and item connects to its five nearest prototypes per channel. Fixed partitions of nearest a, c prototypes define the group, without a conjunction bonus. The original cold group has 276 items, all absent as fitting/validation targets. Confusers enlarge the pool to 776, leaving only 224 outside. Size-matched random pools are unavailable. Models and oracle use identical history exclusions. Oracle eligible top-ten relevance defines group recall on users with a relevant group item and broader recall. For the group-recall reference, an ideal ten-item list can retrieve all relevant items. Logged-relevance NDCG uses a different answer key.

M0/M2/M3 are INTERACTION/SEPARATE/MIXING. All learn Xavier-initialized 128-dimensional ID embeddings, never oracle vectors. M0's two R-GCN layers have widths $128 \to 128 \to 64$ (Schlichtkrull et al., 2018). M2 uses two-layer, width-128 interaction GraphSAGE and prototype-ID context paths (Hamilton et al., 2017). M2/M3 project each of three channels to 21 coordinates, giving effective output width 63 from the nominal setting 64. M3 adds learned sigmoid-gated channel mixing. Both combine backbone and context readouts before dot-product scoring. Held-out target edges are absent from message passing in both orientations.

Fitting and selection. Adam uses 10^{-3} learning rate, 10^{-5} weight decay, batches up to 512, four training-observed negatives per positive excluding user history, pairwise log-sigmoid loss and a 100-epoch cap. Validation NDCG@10 at epoch one and every tenth epoch selects the earliest strict maximum, with patience ten checks. At one-based epoch e, M3 adds to each minibatch's mean pairwise loss

$$\Omega_e = 0.01\,\mathbf{1}\{e \geq 50\} \sum_{t\in\{\text{user,item}\}} \sum_{r,s=1}^{3} \sigma(G^t_{rs}),$$

where G^t are gate logits, initialized to zero, and σ is the sigmoid. All 18 gates, including diagonal entries, contribute without averaging. Mixing operates before the penalty starts. Five of ten original cold M3 checkpoints were selected before epoch 50. Adam weight decay is separate. These models do not match capacity or information.

Original comparison and controls. The original test compares group-only NDCG@10 against hidden recorded interactions with catalogue Recall@10 against preferred group items. Candidates, relevant items and metric change together.

Under the primary criterion, the NDCG winner on the group-only pool ranks last by oracle group recall in 0/5 runs of each block. The secondary all-test-users oracle-NDCG criterion also gives 0/5 throughout. M0 is last in all thirty runs by the group-recall reference and in 29 by all-user oracle NDCG (one viable-warm S2 run is the exception). Six-decimal ties produce neither winner nor last place. Table 4 compares the original lists, ten group slots per list and blind ordering. The cold-block negative has a structural explanation. INTERACTION's cold scores tie in every checkpoint, and even the best single tie order shared by all users scores below the feature-aware models. Only the two feature-aware models exchange places.

The feature models' within-group score AUC against oracle relevance is 0.76–0.78. Logged-relevance NDCG@10 uses a different target. The relabeling test rejects for both feature-aware models in each of the four tested S1 split/training runs ($z = 47.8$–56.8, 997 users). Tests use 1,000 shared cold-column permutations (NumPy `default_rng`), inclusive add-one tails and twice the smaller tail, capped at one. M0's S2 identity-key $p = 0.028$ reflects unusually poor alignment ($z = -2.12$). Using the figures' shared random tie order gives $p = 0.33$. Under one shared key, M0's ten-cold list is exactly the blind list.

Table 4: **Allocation explains the blind win. The three-way pattern recurs.** A: ten original cold checkpoints per model, 997 users, 276 cold items. Ranges span checkpoints/models, not uncertainty. Native paths are as run. The ten-group-slot and blind rows use this table's shared random order. Blind slash pairs are S1/S2. The blind expectation is $10/276 \simeq 0.0362$. Figure 4 uses a second recorded shared random order, with blind group recalls 0.0318/0.0326 and lower global recall than every native feature checkpoint. The original table's scoring route uses a float64 oracle, whereas the figures' route uses a float32 oracle and native top-k. B: ranges span M2/M3 for each new draw/rule, using the figures' shared random order. S1/S2 are paired, not independent draws.

A. Original cold policies: Recall@10

Policy	Group recall	Global recall	Cold slots
M2/M3 native	0.0027–0.0163	0.0347–0.0460	0.09–0.53
M2/M3 ten group slots	0.1799–0.2043	0.1035–0.1185	10
Blind, realized	0.0362/0.0417	0.0221/0.0243	10

B. Two new draws: feature-model ranges and realized blind Recall@10

Draw/rule	Native group	Ten group slots	Blind group	Native broader	Blind broader
43/S1	.0084–.0207	.1842–.1897	.0363	.0418–.0453	.0203
43/S2	.0065–.0163	.2010–.2011	.0377	.0496–.0522	.0215
44/S1	.0062–.0072	.1920–.1989	.0354	.0393–.0399	.0209
44/S2	.0056–.0080	.1883–.2016	.0373	.0450–.0476	.0224

The best possible common tie key. Suppose all users share an eligible cold pool Z of size N, tied model scores and a common total tie order π. Fix nonempty binary relevance $R_u \subseteq Z$, cohort

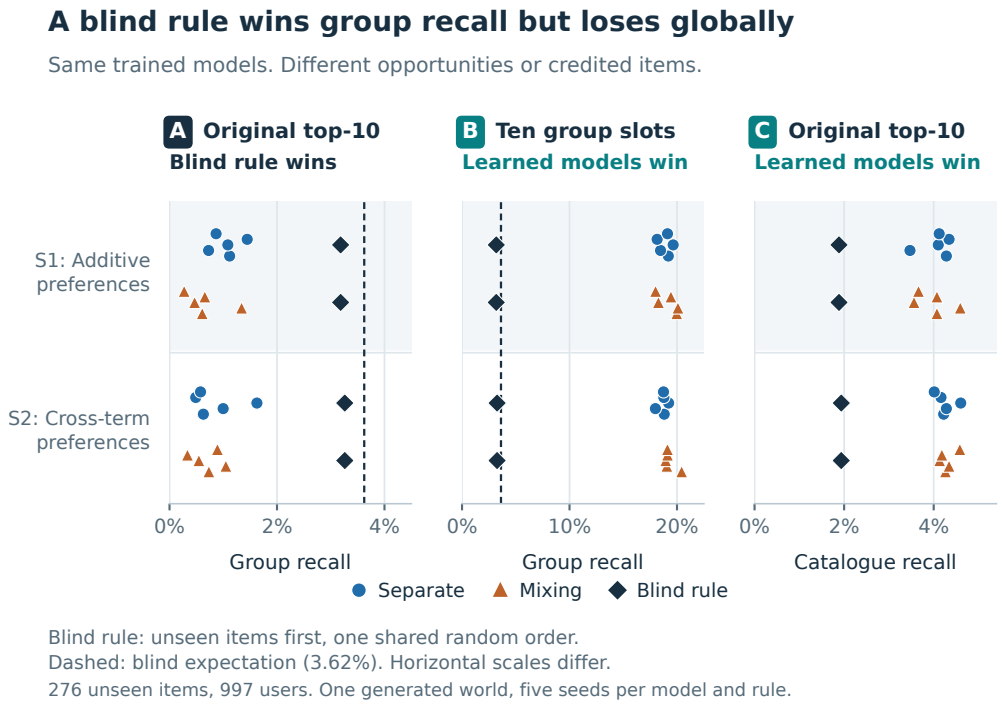

Figure 4: **A blind rule wins group recall but loses on item choice and catalogue recall.** Each circle or triangle shows one training run's mean Recall@10 over the 997 users with a recorded withheld-group interaction for that preference rule. S1 and S2 are the additive and cross-term preference rules in Equation 4. **A:** Score the original top-ten lists on preferred group items. **B:** Score each model's ten highest-ranked group items, without retraining. **C:** Score the original lists on all preferred items in the eligible catalogue. The blind rule recommends only group items in one shared random order. Five split/training runs per model and rule share one generated world. For each preference rule, each panel repeats its fixed blind-rule score on the two model rows. Dashed lines in A and B show its expectation. Vertical offsets only separate marks. Horizontal scales differ. Settings: Appendix A.2. Randomization details are in the reproduction package.

and cutoff. Write $d_r = 1/\log_2(r+1)$ for $r \le \min(k, N)$, zero otherwise, and $a_i = \text{mean}_u[\mathbf{1}\{i \in R_u\}/\text{IDCG}_u@k]$. Exchanging finite sums gives

$$\max_\pi \text{mean}_u \, \text{NDCG}_u(\pi) = \max_\pi \sum_{r=1}^{N} d_r a_{\pi(r)} = \sum_{r=1}^{N} d_r a_{(r)},$$

where $a_{(1)} \ge \cdots \ge a_{(N)}$. For $r < s$ and $a_{\pi(r)} < a_{\pi(s)}$, swapping increases the sum by $(d_r - d_s)(a_{\pi(s)} - a_{\pi(r)}) \ge 0$. Repeated swaps prove the maximum. Even answer-key-informed common keys cannot exceed it, hence neither can user-blind common keys. User-specific keys or differing eligibility are outside this bound. The ceilings 0.046/0.049 are below the minimum feature-model NDCG on the group-only pool 0.150.

Additional realizations and cold decoys. For the two additional draws, all three models are fitted once under paired S1/S2 with a fixed split/initialization setting: twelve fits. Generator/rule/model salts also change batch and negative-sampling streams. The cold sets contain 263/276 items. In draw/rule order 43/S1,43/S2,44/S1,44/S2, cohorts with oracle-relevant group items have 993/996/993/993 users. Broader recall uses full logged cohorts of 993/997/993/993. No separate-factor-support user filter is applied. All feature cases retain the three-way pattern. The original stress-test reversal remains absent. Cold items remain validation candidates. Adding 276 cold decoys from represented combinations breaks equivalence between zero degree and target-group membership. Across ten exploratory feature-model refits, five per model, on 1,000 users (990–995 with a relevant group item), relevant-versus-decoy AUC is 0.731–0.746, versus oracle 0.992 and an

analytic equal-score reference of 0.5. Within-group AUC is 0.739–0.753. Ten-cold group recall is 0.0909–0.1058, against blind expectation $10/552 = 0.0181$. Mean group-versus-decoy class AUC is 0.49–0.51 for the feature models and 0.499 for the oracle.

Why class AUC can average to chance. Fix two nonempty classes independently of the user vector x. Oracle cross-class utility differences are linear in x. With half credit for ties, negation gives $\mathrm{AUC}(-x) = 1 - \mathrm{AUC}(x)$. A centrally symmetric user law therefore gives $\mathbb{E}\mathrm{AUC} = 1/2$ despite individual preferences. A finite cohort need not equal one half. Utility-dependent cohort selection can break that symmetry. Neither this fact nor the discrimination control identifies decomposition.

A.3 Logged pools and allocation

Amazon Electronics contains 374,249 items and 15.9M interactions. Selected users have both constituents separately before splitting. Training retention of that support is not guaranteed. Other users retain training interactions with the three attribute groups. Confusers combine held-brand items with held-category items of top-100 named brands. Same-category brand-`other` items are excluded from the confusers but remain in the catalogue background used for random sampling. Random/confuser size matching precedes per-user history masking, so neither effective size nor attribute sharing is isolated. Table 5 retains the Logitech comparison and all nearest-count decoy signs. Twenty random redraws retain confuser penalties. Random-pool Recall@20 winners are SB-Pop thirteen times and LightGCN seven. Canon has no Recall@20 reversal and zero SB-Pop scores on both matched broad pools. Its NDCG associations differ. Pool-level count profiles do not verify Proposition 4's userwise raw-count condition.

Table 5: **Logitech pool reversals and their popularity control.** A: fixed configurations and positives. Parentheses give pre-mask sizes. B: NDCG@10 for the standard fitted models over twenty redraws greedily matched by nearest training count, without replacement. Matches are not all exact. Closure is the median share of the locked random-minus-confuser gap removed by matching. Matched-minus-confuser is positive on every BPR/LightGCN draw and negative for SB-Pop. The latter's closure exceeds 100%. Panel B retains six decimals so the small gaps and closure percentages remain interpretable.

A. Recall@20 on four candidate pools

	Group only (823)	Random (9,095)	Confusers (9,095)	Catalogue (374,249)
BPR	.356	.267	.122	.036
LightGCN	.381	.286	.127	.022
SB-Pop	.358	.298	.152	.000

B. NDCG@10 after approximate popularity matching

	Confuser	Random	Matched median	Matched range	Closure
SB-Pop	.018771	.109382	.018148	.018148–.018148	100.7%
BPR	.037304	.104188	.049053	.048592–.049366	82.4%
LightGCN	.023530	.115663	.037189	.037059–.037304	85.2%

The fixed descending-score/ascending-ID order certifies reversals for all three Logitech pairs and eight Sony pair/pool records (six tie-free). Three Amazon cutoff cases depend on scoring path, although no verdict relies on them. We do not claim backend-independent strictness. These certificates are descriptive, not significance tests or predictions for every pool. Figure 3 uses stable-order, unrounded per-user replacements, not counts of winners at a common group size. With twenty group slots, LightGCN wins for 559 Logitech users, BPR for 380, with 4,065 ties.

LightGCN has higher *mean* within-group recall at cutoffs one to twenty, but each model wins for some users and most tie at twenty. These averages do not determine the ordering component at the models' own slot counts.

Interpreting the pool controls. The three-model order completely reverses between either matched-size pool and the catalogue. Under Corollary 1's fixed-order conditions, these reversals

refute itemwise dominance. The gap remaining after popularity matching keeps its sign across redraws, but does not by itself identify a composition effect. The Logitech attribute combination remains represented in training, so these comparisons explain pool and allocation effects rather than generalization to a model-novel attribute combination.

Paired-user uncertainty. Figure 3 retains all 5,004 aligned users, including zero-slot and zero-hit rows. From each user's saved within-group recall curves and native counts, we reconstruct the four crossings of the form in Equation 1. We draw 20,000 bootstrap samples of 5,004 users with replacement, using identical indices for all four crossings. Each sample recomputes the gap, both averaged components and all swap-path terms. Pointwise 95% intervals use the 2.5th and 97.5th percentiles with linear interpolation. The accounting identities hold userwise and in every resample. These intervals condition on the saved checkpoints, catalogue, relevance and split, not training seeds or new datasets. The repository retains all eleven estimates, intervals and the fixed resampling specification.

A.4 The near-cold movie control

Comedy–UK contains 266 of 8,155 films and is the largest of 443 constructible groups. Removing qualified users' group interactions leaves 56 films with training interactions and 210 without. Three of the latter occur in validation. Across 1,926 test users, none of the six original fitted baselines places a training-cold film in any catalogue top-twenty list. The same absence holds for the 207 films without training or validation interactions. Broader-pool NDCG@10 is zero for all six. LightGCN's 11,044 and ItemKNN's 9,723 top-twenty hits on the group-only pool are all residually warm films. These allocation and hit-composition facts leave selector symmetry untested. The complete secondary warm-group and relabeling studies, including contrary outcomes and tie-path differences, remain together in the extended record.

A.5 Released-checkpoint application

We compare the released models' item choices at the same group positions, then bound the reverse comparison when TIGER's ranking is incomplete.

Pinned released SASRec/TIGER weights and public defaults reproduce three category NDCG@10 means per model/domain to four decimals (Ding et al., 2026). Beauty/Steam have 22,363/47,761 test users, 12,101/10,971 valid item IDs, and 18,170/29,354 generalization-target users, retaining misses and users with no group recommendations.

Groups reproduce the released categorizer without test targets: union one/two-edge reachability from the last h history items in the undirected lag-at-most-h training graph, excluding the last item's recorded one-step successors, $h = \min(4, |\text{history}|)$. The cohort contains all users whose target has the published generalization category. These are two domains of one study.

Histories stay eligible. SASRec preserves its native prefix, then orders logits decreasingly with one shared PCG64 tie key, not a recovered backend order. Its two/three off-prefix target ties cannot change hybrid hits. TIGER has 44 invalid slots in Beauty and five in Steam, with no repeated valid slots on these cohorts.

Position-preserving NDCG accounting. For fixed nonempty binary $R_u \subseteq C_u$, complete within-group order O and the group's positions $P = (p_1 < \cdots < p_l)$ among the original top-k slots, define

$$H_u(O, P) = \frac{1}{\mathrm{IDCG}_u@k} \sum_{j=1}^{l} \frac{\mathbf{1}\{O_j \in R_u\}}{\log_2(p_j + 1)}.$$

Positions mark first valid, distinct group items. Invalid, repeated and non-group slots retain their uncredited gaps. SASRec's preserved prefix gives native score $s_u = H_u(O_{S,u}, P_{S,u})$. With $b_u = H_u(O_{S,u}, P_{T,u})$ and TIGER native t_u, cancellation gives

$$\mathrm{mean}_u(t_u - s_u) = \underbrace{\mathrm{mean}_u(b_u - s_u)}_{\text{placement substitution}} + \underbrace{\mathrm{mean}_u(t_u - b_u)}_{\text{item choice}}.$$

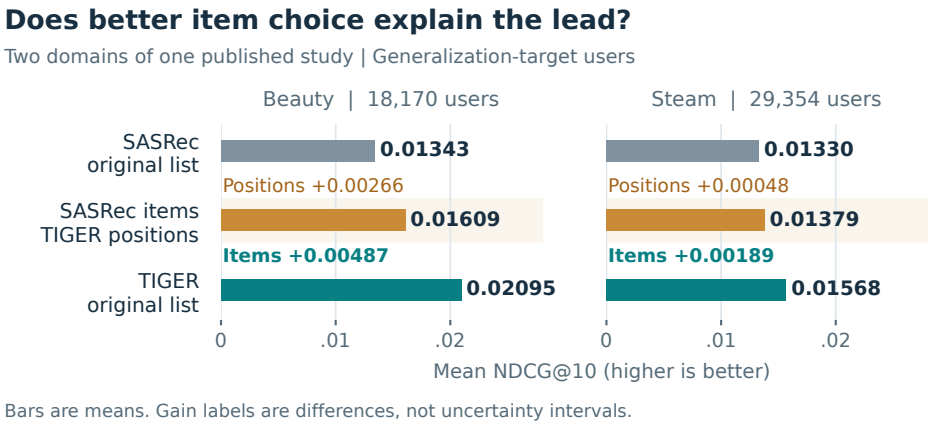

Figure 5: **Matching opportunities does not erase TIGER's advantage.** Each panel fixes its generalization-target cohort. The middle list fills TIGER's group positions with SASRec's choices. Top to middle changes placement. Middle to bottom changes choice at those positions. Both panels use fixed released checkpoints from two domains of one study. Differences use unrounded means. Methods and intervals: Appendix A.5.

This single sequence supplies neither a symmetric crossed comparison nor a causal architecture decomposition. It infers no complete TIGER order.

Table 6: **The choice advantage survives both native placement references.** Upper rows: mean NDCG@10 differences and 95% paired-user percentile intervals (2,000 draws), conditional on checkpoints, items, split and declared order, not retraining or dataset uncertainty. Lower rows: formal reverse-comparison bounds, not confidence intervals. The top-fifty sensitivity was run only for Steam. Beauty uses its recorded top-ten prefix. Figure 5 gives the three policy means.

Contrast	Beauty	Steam
Placement substitution	.00266 [.00219,.00315]	.00048 [.00010,.00088]
Item choice at fixed positions	.00487 [.00306,.00671]	.00189 [.00071,.00297]
Reverse comparison: formal completion bounds		
Top-ten prefix	[.004611,.031087]	[.000288,.084360]
Top-fifty prefix	n/a	[.001082,.001181]

The matched-position hybrid is post-processing of saved item lists, not a decoder-token intervention. Neither model is required to be able to generate it.

Sharp completion bounds. We use standard position-weight and partial-ranking ideas (Singh & Joachims, 2018; Moffat et al., 2025). For one relevant target, fix TIGER's distinct group prefix of length $d \leq N$ in the same eligible group of size N, and SASRec's valid, distinct group positions $p_1 < \cdots < p_q$. Put $w(r) = 1/\log_2(p_r + 1)$ for $r \leq q$, zero otherwise. An observed target at group rank r scores $w(r)$. An unobserved target can occupy any rank $d + 1, \ldots, N$ in a formal prefix-preserving completion:

$$w(N) \leq x \leq w(d + 1).$$

Placing the target last or immediately after the prefix attains the endpoints. Subtracting native SASRec scores and averaging gives sharp fixed-cohort bounds over independent per-user completions. These bound missing ranking information, not sampling or recovered decoder rankings. Shared decoder constraints may narrow feasibility. Neither userwise dominance nor arbitrary-placement robustness follows. Steam's first fifty positions from the same beam output tighten the bound, preserving native top-ten scores and SASRec positions. The top-ten lower bound was already positive.

Repeat-aware item-neighbour control. We compare the learned models with a repeat-aware item-co-occurrence baseline at the same group positions. The baseline sums cosine similarities to distinct history items. Similarities use binary training incidence and zero diagonal. Training excludes the final two events, and query history excludes only the last. U sorts scores with an independent tie key. D moves unseen items before seen items, preserving each part's order. Neither uses the test answer. Both fill TIGER's original group positions and retain all users, including 43/5,116 with no such positions.

Table 7: **TIGER retains an advantage over the repeat-aware neighbour.** Paired mean NDCG@10 differences at TIGER group positions. The 95% pointwise paired-user percentile intervals use 10,000 fixed-input draws on exposed data, excluding retraining, selection and multiplicity uncertainty. Beauty SASRec–D is inconclusive.

Contrast	Beauty	Steam
TIGER – D	+0.003482 [+0.001535, +0.005450]	+0.004451 [+0.003351, +0.005585]
SASRec-at-TIGER – D	-0.001384 [-0.003239, +0.000400]	+0.002562 [+0.001381, +0.003704]
D – U	+0.006108 [+0.005358, +0.006860]	+0.000687 [+0.000397, +0.000980]

D cannot worsen unseen-target ranks but can hurt repeats. On Beauty, 370 users improve under D relative to U. On Steam, 128 improve and 20 worsen. Beauty has no repeat targets. Steam's 136 repeat targets score zero under D because unseen items fill every group position. They contribute descriptively 12%/39% of the TIGER/SASRec minus D means, without subgroup intervals. Beauty's SASRec–D interval establishes neither equivalence nor disadvantage.

The neighbour assigns zero similarity to the target for 9,553 Beauty users and 1,266 Steam users. The reproduction record retains all contrasts and historical-control reconstruction checks.

A.6 Frozen item-choice correction on additional Steam users

Cohort and outcome. Users come from the same Steam archive and period as development, with original zero-based post-core insertion-order index congruent to one modulo seven rather than zero. The history-only eligibility rule excludes 971 of 47,761 candidates because their histories contain items outside the frozen vocabulary. All 46,790 eligible users are scored, including 159 whose next item is outside the vocabulary, scored as misses. A hit at original position $j \leq 10$ scores $1/\log_2(j+1)$, and a miss scores zero. Invalid or repeated native slots remain uncredited gaps. The source-defined users do not overlap correction development. The two residue classes each contain 47,761 users because the archive has 334,325 post-core users. According to the checkpoint documentation and default source pipeline, only residue-zero users enter fitting, validation and checkpoint selection. Historical processed-cache provenance was not independently certified. The earlier structural core filter used the full archive. This is not temporal or independent-source validation.

Correction and selection. Each user has three target-independent groups: the last item's recorded one-step successors M_u, the history-reachable group G_u of Appendix A.5, and remaining catalogue items U_u. The graph and binary-incidence cosine similarities use only the original training support, omitting each original user's final two events. Query histories do not refit this support.

We use weighted reciprocal-rank fusion (Cormack et al., 2009). Let $T_u(i)$, $S_u(i)$ and $D_u(i)$ be reciprocal ranks $1/(60+r)$, with one-based rank r. TIGER's order concatenates its native ten-item and recorded fifty-item lists, removing invalid and repeated items, with $T_u(i) = 0$ for unrevealed items. SASRec preserves its native prefix before completing the catalogue order. The similarity order sums cosine similarities to distinct history items, with zero diagonal, then places items absent from the history before seen items while preserving both internal orders. SASRec tail and similarity ties use one shared PCG64 order. The correction orders by

$$F_u(i) = \begin{cases} T_u(i) + 0.5S_u(i), & i \in M_u, \\ T_u(i) + 0.5S_u(i) + 0.5D_u(i), & i \in G_u, \\ S_u(i) + D_u(i), & i \in U_u. \end{cases}$$

Ranks are computed before group filtering. Fusion ties use complete SASRec rank. Each group's prefix fills only that group's prescribed positions.

Selection maximized mean full-list NDCG@10 at native TIGER group positions on 23,880 calibration users from the previously exposed 47,761-user Steam development population. A fixed PCG64 permutation divided that population, with the first half used for calibration. The other 23,881 users supplied development assessment, not untouched confirmation. For each group, selection chose between keeping its incumbent order and the 26 nonzero weight triples in $\{0, 0.5, 1\}^3$, represented by 19 positive-scale classes in the exact-tie implementation. Selection sums the group's contribution over all calibration users. The best fusion replaces the incumbent only for a strict improvement, with the first maximizing option in a fixed enumeration resolving fusion ties.

The selected global control uses $T + 0.5S$ in every group. Its selection family allowed similarity and repeat-aware terms, but its chosen rule has neither. It maximized the same calibration objective over 191 options: keep, 19 fusion classes each with raw or seen-demoted cosine, and eight repeat-handling variants of each raw-cosine class. These variants add $\rho\mathbf{1}\{i \text{ is in the history}\}/61$ to the canonical-scale fusion score, with $\rho \in \{-2, -1, -0.5, 0.5, 1, 2\}$, or place unseen or seen items first. A user-context control uses $T + S$ in the middle of three development-defined bins of history repeat share and $T + 0.5S$ otherwise. Repeat share is $1 - |\text{distinct history}|/|\text{history}|$. The reproduction configuration preserves the exact cutpoints, float64 calculation and boundary convention. All rules and tests were fixed before scoring additional-cohort outcomes. The repository gives the development selection grid and exact selected rules. These known fusion ingredients are an application, not a new fusion mechanism. This test compares with the selected global rule, not every possible global combination, and does not isolate the necessity of group conditioning.

Four placements. We test native TIGER positions, native SASRec positions, TIGER's group counts arranged as M, G, U blocks over its non-gap slots, and TIGER positions with up to two rightmost valid non-G slots changed to G when capacity permits. Within each placement, all rules share group positions and gaps. The incumbent is native SASRec at SASRec positions. Elsewhere it uses TIGER's revealed order followed by SASRec when needed, reproducing native TIGER at its own positions. The other two incumbents are constructed lists. The global comparator is also constrained to these positions, not its own unconstrained top ten. Group positions do not fix seen-item positions.

Table 8: **The correction retains its advantage under four placements of one cohort.** All rows use the same 46,790 Steam users. Means are NDCG@10. Differences use unrounded means. Intervals for correction minus global are Bonferroni-2 primary (first row) or gated Bonferroni-11 secondary (other rows). They condition on the saved models and cohort, not training seeds or a new source.

Placement	Incumbent	Global	Correction	Correction − global
TIGER	.154304	.155220	.156538	$+.001318$ $[+.000870, +.001767]$
SASRec	.150332	.155226	.156463	$+.001236$ $[+.000674, +.001798]$
Group blocks	.154347	.155352	.156278	$+.000926$ $[+.000459, +.001393]$
Extra G slots	.151829	.152701	.154200	$+.001499$ $[+.000911, +.002086]$

Uncertainty and tradeoffs. For paired differences d_u, intervals are $\bar{d} \pm z s_d/\sqrt{n}$, with sample standard deviation s_d. The two primaries compare correction against global and native TIGER at TIGER positions, using Bonferroni-2 intervals. The latter difference is $+0.002233$ with interval $[0.001458, 0.003009]$. Both lower endpoints exceed zero. All eleven prespecified secondary positive-effect tests pass a gated Bonferroni-11 family: correction minus global and incumbent at the other three placements, correction minus user-context control at all four, and global minus native TIGER at TIGER positions. The last passes narrowly, $+0.000915$ $[0.000023, 0.001808]$. This does not give joint 95% coverage for all thirteen intervals. All contrasts are retained in the repository.

Relative to the global control at TIGER positions, 26,576 lists change (56.80% of users) but only 979 scores do (2.09% of users, 3.68% of changed lists): 606 improve and 373 worsen. The gain is carried by the 28,985 answers in G_u (mean $+0.002128$). The M_u rule equals the global rule, and the 283 answers in U_u show no score difference. Gains on the 40,690 non-repeat answers average $+0.001847$, while the 6,100 recorded repeat answers lose 0.002208 on average. Repeat answers also lose against native TIGER, by 0.003280 on average. These outcome-defined strata do not identify future beneficiaries. The extra-G placement has a lower mean than TIGER's placement under all

three displayed rules, despite a positive correction effect within it. This is a descriptive, untested comparison across placements. Four placements are not four independent replications.

Development did not establish additional benefit over the global control on Beauty. The earlier development criterion of a residual point estimate at least 1% of incumbent NDCG remains unmet. On the additional Steam cohort, the residual is 0.85% of incumbent NDCG. The descriptive interval for its margin from the 1% reference crosses zero, so it establishes the residual neither above nor below that reference. An old-development diagnostic additionally fixing seen-item positions was inconclusive. None of these is an equivalence result or a causal explanation of the gain. Both model outputs and graph/similarity support are required. The repository records batch-pipeline timings for inference and list construction, not marginal online latency.

A.7 Audit procedure and diagnostic controls

Five-step audit procedure. To investigate why one recommender scores higher and whether a targeted correction helps, we propose the following five-step procedure. It connects the audit to additional checks and to evaluating a correction on separate users.

1. Define the comparison. Define groups without test answers and record the evaluation users, eligible and relevant items, and fitting and selection inputs. For a compositional claim, identify the attributes or learned factors and whose training data lack their combination.

2. Compare choices at matched opportunities. For recall, compare each model's group choices at both models' per-user counts and report the original gap and its allocation and ordering components (Section 3.1).

For NDCG, preserve the positions and discounts of valid, distinct group items (Section 6.3).

An advantage that survives at matched positions supports better choices there. If it disappears, placement can explain the original advantage.

With no group items recommended, compare a feasible common count using deeper rankings. Missing ranks require completion bounds or an unresolved comparison. The reusable diagnostic described below provides the evaluator and separate single-target completion-bound routines.

3. Add an appropriate chance comparison. Give an explicit random-order control the same per-user group counts. It needs no learner symmetry, and its realized score can differ from its expectation.

A cold-item learner null or relabeling test requires the conditions in Appendices B and B.2, respectively. The further audit checks below cover warm and near-cold groups.

4. Vary candidates while keeping relevant items fixed. Keep rankings, relevance, cutoffs, normalizers and users fixed. Use the pool controls described under further audit checks below. A reversal shows dependence on the pool.

The stronger conclusion about dominance requires Corollary 1's conditions.

5. Test a correction on separate users. Select an item-choice correction on one set of users, with controls such as one rule shared by all groups. Freeze the rules and tests, then evaluate separate users at fixed group positions. Grade complete lists, including positions outside the audited group, and report subgroup losses alongside the mean.

Section 7 demonstrates the sequence.

Reusable diagnostic and prior controls. The NumPy API and examples in `src/snd/audit/` take fixed-cohort eligibility, relevance and group arrays, with complete orders for recall or an observed top-k list and complete comparator for NDCG. Outputs are native/matched policies and components. The separate completion-bound routines for single-target NDCG@10 are documented in `experiments/published/README.md`. Mixed relevance permits only group-contribution

recall accounting. Array checks certify neither target independence, information access nor symmetry. Repeat/explore controls and quotas are established (Li et al., 2023a;b; 2024; Liu et al., 2025). Our substitutions instead retain recorded per-user opportunities rather than optimizing budgets or discarding repeats. Cold-item learner nulls additionally need complete-procedure symmetry.

Further audit checks. Groups containing warm items need a separately justified learner null. For near-cold groups, specify whether the fixed group count includes only training-cold items. Declare the statistic and tail in advance when applying the shared-relabeling test in Appendix B.2. For candidate-pool checks, match confuser and random pools in size before history masking and report effective sizes, variation across random draws and scores near zero (Appendix A.3). Label exploratory analyses and distinguish stored summaries from replays.

Chance references and pool checks. How high can group recall be without choosing items according to a user's preferences? Fix $N > 0$ eligible cold items containing the user's fixed, nonempty relevant set. If the ranked-output law, including ties, is invariant under every permutation of these items, expected recall conditional on selecting ℓ of them is ℓ/N, for counts with positive probability.

More group positions raise expected recall without better item choices. Proposition 6 in Appendix B.4 gives the full benchmark and its attainable upper bound.

Using this expectation as a learner null needs justified ranked-output symmetry, not merely missing training interactions. Appendix B gives sufficient fitting, selection and tie conditions. Symmetry of model selection in the recorded runs remains an unverified hypothesis (Appendix A.1).

Appendix B.2 gives a separate test of alignment with relevance and its conditions. It renames unseen items with one permutation shared across users.

The allocation and ordering audit itself needs no symmetry assumption: it compares saved recommendations.

Adding the same offset to every score in a group can change allocation without changing within-group order (Remark 1). Ordering alone therefore cannot capture all compositional behavior.

Candidate-pool implications. Changing competing candidates, with rankings and relevance fixed, tests whether a comparison depends on its pool.

We say model A *itemwise dominates* B when, for every user and relevant item, every item ranked above it under A is also ranked above it under B.

Appendix B.3 gives the conditions under which reversals rule out either direction of dominance. It also states when a raw-popularity ranker must have zero Recall@k and NDCG@k. That result concerns raw interaction counts, not the sampled-batch counter SB-POP.

B MATHEMATICAL SUPPLEMENT

We first state when cold-item identities admit a random-order reference and a calibrated relabeling test. We then prove the candidate-pool and score-accounting results.

Definition 4 (complete-input item symmetry)**.** Let O contain every input read by fitting or checkpoint selection, including validation and side information. Let Z be items absent from these records and let the permutation group G permute Z while fixing O and all items outside Z. *Score symmetry* is G-invariance, conditional on O, of the selected score field jointly across users. *Ranked-output symmetry* is the corresponding invariance of complete rankings, including tie handling. Global item symmetry means that relabeling all inputs relabels the joint score-field law.

Users and items are finite. The action is $(gX)_{u,i} = X_{u,g^{-1}i}$ on scores and relabels identities at fixed positions on rankings. One g acts across all users and, for a pooled statistic, all executions. The fixed evaluation object y includes relevance, eligibility, pools, cohorts and grading conventions. These must be invariant or transform with y so that $T(gX, gy) = T(X, y)$. All laws below condition on O.

Unless stated otherwise, probability laws condition on O, not a realized checkpoint. $\stackrel{d}{=}$ denotes equality in distribution.

Conditional cold-item benchmark. Equivariant fitting and an invariant selector suffice for score symmetry (Lemma 1). Historical selector symmetry remains an unverified hypothesis (Appendix A.1). An equivariant ranking kernel then gives ranked-output symmetry. Fixed identity keys on equal scores do not: a tie-averaged metric can instead use the score-field route. Eligibility, pools and cohorts must be invariant or transform with the answer key.

Proposition 2 (conditional cold-item symmetry)**.** *Fix the complete input O and an answer key and evaluation construction y. Let X be either the selected score field or the complete ranked output, with $gX \stackrel{d}{=} X$ given O for every $g \in G$. For any statistic obeying $T(gX, gy) = T(X, y)$,*

$$T(X, gy) \stackrel{d}{=} T(X, y) \qquad \textit{conditional on } O.$$

The statement holds per learner. Pooling executions requires their joint *G-invariance, not merely invariant marginals.*

Under all eligible cold permutations, fully cold Recall and NDCG have uniform-random expectations: use the ranked-output route or tie-averaged metrics under the score-field route. This is not every realized score or a common aggregate law across learners. On mixed pools, invariant allocation conditioning preserves symmetry without equating cold placement.

This symmetry concerns anonymous IDs, not unseen items with informative descriptions. Retaining training interactions for the declared attribute or factor combination removes its model novelty. Neither a cold nor a warm construction alone establishes learned-factor composition. Novelty under other learned factors remains possible.

Testing alignment without assigning information to item IDs. Relabel unseen answer-key identities with one permutation shared across users. Under Proposition 2's conditions, the resulting alignment test has a per-test false-positive bound. The bound is conditional on the chosen object's orbit, meaning the outputs reachable by permitted renamings, rather than on its realized identity assignment. Appendix B.2's Proposition 3 specifies inclusive tails, two-sided doubling and the add-one Monte Carlo rule. Rejection can reflect features, selection, leakage, identity ties or a false positive, not necessarily composition. Non-rejection proves neither symmetry nor incapability. Two-sided rejection can indicate unusually poor alignment.

B.1 Fitting, selection and cold-identity symmetry

Lemma 1 (conditional equivariance of fitting)**.** *For a finite training horizon, separate item-indexed state, including optimizer state, from label-free shared state. Assume under simultaneous relabeling of inputs and state: (a) The conditional joint initial-state law on gO is the g-pushforward of its law on O (for example, a data-independent joint law with i.i.d. item blocks independent of shared state). (b) The loss is invariant and the score readout is equivariant. (c) Given every history, the conditional law of each next random draw transforms with that history, including minibatches, negatives, dropout and state-dependent draws. (d) Updates commute with relabeling, including optimizer state, hyperparameters, coordinate operations and any derivative-selection rule. Then the joint fitting trajectory and score readouts are equivariant. For $gO = O$, an invariant checkpoint selector preserves score symmetry, and an equivariant ranking kernel preserves ranked-output symmetry.*

Proof. Couple runs on O and gO by relabeling the initial item blocks and keeping shared state fixed. Condition (a) gives the correct initial law. If their histories are relabelings, (c) couples the next draws in the same way. For a differentiable invariant loss, differentiating its invariance identity permutes the item gradients and leaves the shared gradient unchanged. Together with (d), the next states are therefore relabelings. Induction gives the joint trajectory law, and (b) gives its score readouts. An invariant selector chooses the same checkpoint index on coupled paths. Finally, a ranking kernel satisfying $K(gs, gA) = K(s, A)$ for ranking events A carries invariant score laws to invariant ranking laws. □

This includes invariant stopping within the horizon and invariant randomized selection whose randomness is in the history. Condition (c) concerns conditional laws, not marginal sampling frequencies. Condition (d) also covers nonsmooth derivative choices. These are intended-arithmetic conditions, not a certification of floating-point execution or historical selectors.

The selector and ties require their own checks. A sufficient rank-validation condition is that, at every compared checkpoint, each validation positive has a finite score distinct from *every other eligible item's* score. Its rank is then one plus the number of strictly higher scores, unchanged by cold relabeling when validation relevance and eligibility are invariant. Invariant aggregation, rounding and stopping then preserve selection. With ties, the criterion itself needs an equivariant or tie-averaged construction. This condition has not been established for the historical trajectories. An all-zero score field with a fixed identity tie order is score-symmetric but not ranked-output symmetric. An equivariant random tie kernel, or its expectation inside the statistic, supplies the respective valid route.

Proof of Proposition 2. For the invariant object X of the chosen route and any fixed $g \in G$,

$$T(X, gy) = T(g^{-1}X, y) \stackrel{d}{=} T(X, y).$$

If G contains all permutations of the eligible cold set, it acts transitively on its $N!$ relative orders, so ranked-output symmetry makes each equally likely. A fully cold pool therefore has uniform-ranking Recall and NDCG expectations. Under score symmetry, apply an equivariant tie kernel and then average the metric conditional on scores to obtain the same tie-averaged expectations.

Conditioning on an invariant summary preserves invariance: its level sets are unions of orbits. Examples are warm scores and the unlabeled cold-score multiset, or cold-occupied ranking positions. The summaries and cross-user dependence may differ between learners, so neither mixed cold placement nor aggregate laws must agree. Pooling executions requires their joint invariance under the shared action. Invariant marginals alone do not suffice. Finally, a training occurrence of the declared combination removes model novelty by Definition 1, without excluding novelty under other latent maps. □

B.2 A CALIBRATED SHARED-RELABELING TEST

The test asks whether the observed alignment with relevant items is unusual under one permitted renaming shared across users.

Proposition 3 (shared-relabeling test)**.** *Under Proposition 2's invariant-object and statistic conditions, let $Q = \mathrm{orb}_G(X)$ and define inclusive tails*

$$p_{\mathrm{hi}} = \frac{1}{|G|}\sum_{g\in G} \mathbf{1}\{T(X, gy) \geq T(X, y)\}, \qquad p_{\mathrm{lo}} = \frac{1}{|G|}\sum_{g\in G} \mathbf{1}\{T(X, gy) \leq T(X, y)\}.$$

Each tail obeys $\Pr(p \leq \alpha \mid O, Q) \leq \alpha$ for $0 \leq \alpha \leq 1$, as does $\min\{1, 2\min(p_{\mathrm{hi}}, p_{\mathrm{lo}})\}$. For B independent uniform $g_b \in G$, independent of X, replace a tail by

$$\widehat{p}_{\mathrm{hi}} = \frac{1 + \sum_{b=1}^{B} \mathbf{1}\{T(X, g_b y) \geq T(X, y)\}}{B + 1}$$

(reverse the inequality for $\widehat{p}_{\mathrm{lo}}$). The same level bound and two-sided rule hold jointly over learner and permutation randomness. The bounds also condition on any invariant summary of X, but not generally on its realized identity-bearing value. They are per test.

Proof. Conditional on Q, invariance makes X uniform on that finite orbit. This also holds for real score arrays: each finite orbit has a measurable lexicographically least representative. Conditional invariance and transitivity give uniformity for almost every representative. The map $g \mapsto g^{-1}x$ has equal-size stabilizer fibers onto Q, hence

$$p_{\mathrm{hi}}(x, y) = \frac{1}{|Q|}\sum_{x'\in Q} \mathbf{1}\{T(x', y) \geq T(x, y)\}.$$

Sort these values in descending order. Each tied group receives its largest rank, so at most an α fraction of orbit points has inclusive upper rank at most $\alpha|Q|$. Uniformity proves the upper-tail bound. Ascending order proves the lower tail. For $\alpha < 1$, rejection by the doubled minimum requires a tail at most $\alpha/2$, so the union bound proves two-sided validity. The case $\alpha = 1$ is immediate. Averaging over orbits proves the invariant-summary claim.

For Monte Carlo, conditional on Q and $X = x$, the arrays $g_b^{-1}x$ are i.i.d. uniform on Q, with a distribution independent of x. This makes $(X, g_1^{-1}X, \ldots, g_B^{-1}X)$ conditionally i.i.d. uniform on Q. The add-one value is the observed statistic's inclusive upper rank among these $B+1$ exchangeable values, divided by $B+1$. The same rank argument proves its bound, including ties. The leading one counts the observation itself. Lower tails and doubling follow as above. □

Use the score field's orbit for a score-field statistic, never the orbit of an identity-broken ranking. Choosing statistics or checkpoints after inspecting results needs a further multiple-testing argument. Rejection can reflect features, selection, leakage, identity ties or a false positive. Non-rejection establishes neither symmetry nor incapability.

B.3 Candidate pools and dominance

Proposition 4 (eligible popularity floor)**.** *In an eligible pool P after history masking, fix nonempty $R \subseteq P$ and $1 \le k \le |P|$. If at least k items in $P \setminus R$ have strictly more recorded training interactions than every item in R, descending raw popularity gives zero Recall@k and NDCG@k, regardless of ties.*

Proof. Those k irrelevant items precede every relevant item, so the top k contains no positive. Recall and DCG vanish. Binary IDCG is positive. □

Definition 5 (itemwise dominance)**.** Fix each learner's total catalogue order for each user, including a pool-independent tie rule. Learner A itemwise dominates B for fixed R_u when $\{j : j \succ_A i\} \subseteq \{j : j \succ_B i\}$ for every u and $i \in R_u$, where $j \succ_A i$ means j precedes i under A.

Proposition 5 (pool-invariance under dominance)**.** *If A itemwise dominates B, restricting these same orders to any common eligible pools containing the same relevance gives A at least B's binary Recall@k and NDCG@k, per user and under common nonnegative user weights. Relevance, cutoffs, normalizers, weights and grading cohort are fixed.*

Corollary 1 (reversals imply failure of dominance)**.** *Under these conditions, strict opposite aggregate comparisons on two pools rule out either direction of itemwise dominance.*

Proof. Intersect each predecessor inclusion with the eligible pool P:

$$r_A^P(i) = 1 + |\{j \in P : j \succ_A i\}| \le 1 + |\{j \in P : j \succ_B i\}| = r_B^P(i).$$

Each positive's contributions $\mathbf{1}\{r \le k\}$ and $\mathbf{1}\{r \le k\}/\log_2(r+1)$ are non-increasing with r. Summing and dividing by the same positive normalizers proves the per-user inequalities. Nonnegative common weights preserve them. Either dominance direction would therefore contradict one strict reversed comparison. Dominance is sufficient, not necessary, for absence of reversals. □

B.4 Recall accounting, weights and placement

Proof of Proposition 1. The group items in the first $\min(k, |E_u|)$ positions form exactly a length-L_u prefix of the same total order restricted to C_u. Both recalls count its relevant items and divide by $|R_u|$. □

Adding Equations 2 and 3 cancels the cross terms, giving $\mathcal{A} + \mathcal{D} = q_{AA} - q_{BB}$. All four entries use the same cohort and grading convention and preserve each user's order–allocation pairing. This is policy accounting. It implies neither causal architecture effects nor a warm-group chance law.

Proposition 6 (occupancy identity)**.** *Fix a user's eligible pool P, a cold set $Z \subseteq P$ of size $N > 0$, nonempty fixed relevance $R_u \subseteq Z$, and $1 \le k \le |P|$. Suppose the ranked-output law, ties included,*

is invariant under every permutation of Z. If L_u counts cold items among the first k ranks, then, for every value ℓ of positive probability,

$$\mathbb{E}[\mathrm{Recall}_u@k \mid L_u = \ell] = \frac{\ell}{N}, \qquad \mathbb{E}[\mathrm{Recall}_u@k] = \frac{\mathbb{E}[L_u]}{N} \le \frac{\min(k, N)}{N}.$$

The bound is attained by filling all available slots with cold items and ordering them symmetrically. Additional invariant allocation conditioning is allowed. Conditioning on the full identity-bearing ranking is not.

Proof of Proposition 6. Conditional on $L = \ell$ of positive probability, symmetry makes the N cold inclusion probabilities equal. Their sum is ℓ, so each is ℓ/N. Summing over fixed nonempty $R \subseteq Z$ and dividing by $|R|$ gives expected recall ℓ/N. The tower rule and $L \le \min(k, N)$ give the unconditional identity and bound. Cold-first ranking with an exchangeable cold order attains it. Further invariant conditioning preserves the argument. Zero-degree-first is this construction only when its eligible zero-degree block is exactly Z. □

Proposition 7 (weighted and mixed cold contributions)**.** *Under the same ranked-output symmetry, let B be the cold-occupied positions among the first k ranks, $g_i \ge 0$ fixed cold gains and $d_j \ge 0$ fixed discounts. Writing $\iota(j)$ for the item at position j,*

$$\mathbb{E}\left[\sum_{j \in B} d_j g_{\iota(j)} \,\middle|\, B\right] = \frac{\sum_{i \in Z} g_i}{N} \sum_{j \in B} d_j.$$

Division by a fixed positive relabeling-invariant normalizer gives the cold NDCG contribution. For fixed binary relevance with $R \cap Z \neq \varnothing$ that may also include warm items, the expected cold recall contribution given $L = \ell$ is $\ell|R \cap Z|/(N|R|)$. Warm contributions are separate. Additional invariant warm-ranking or allocation conditioning is allowed.

Proof. Conditional on B, every cold identity has probability $1/N$ at each $j \in B$. Linearity gives the identity. For mixed recall, each cold positive has inclusion probability ℓ/N. A random invariant normalizer must also be conditioned on before pulling it outside the expectation. □

NDCG depends on the occupied positions, not only their count: a single positive at position one or two has NDCG 1 or $1/\log_2 3$.

Remark 1 (within-group order and group placement). For a finite eligible catalogue with finite real scores and a fixed shift-invariant tie rule, adding λ to every score in a target group C preserves its within-group order. If C and its complement each contain at least k items, suitable shifts make occupancy k or 0.

Proof. Within-group differences do not change. Choose λ above all outside-group-minus-inside-group score differences, or below all of them, placing the entire group above or below its complement. This uses exact real arithmetic. Within-group order alone cannot measure group placement or locate every possible compositional behavior. □

A cost of forcing allocation. Take 100 cold items with one positive and ten warm items with nine positives. A model ranks the cold positive first within its group but recommends the ten warm items. Group recall is zero and global recall is $9/10$. Ten uniformly chosen cold items have expected group recall $10/100$. Forcing the model's ten slots into that group gives perfect group recall but global recall $1/10$. Equal allocation can diagnose ordering while reducing utility.