From Head to Tail: LLM-Augmented Sequential Recommendation and Trade-off Optimization

Anonymous ACL submission

Abstract

Recommender systems exhibit popularity bias, often under-serving long-tail and cold-start items with sparse interaction data. We propose a group-specific data augmentation approach that leverages LLMs to generate synthetic interactions, formulated as a constrained optimization problem, maximizing group-specific NDCG@20 and coverage while bounding overall NDCG@20 degradation within an acceptable threshold. Items are partitioned by interaction frequency to find the optimal augmentation strategy for each group. We further study combining the augmentation across groups simultaneously and find that while individual optimal strategies navigate the multi-group augmentation design, they are not simply additive. We further ablate on the LLM-based augmentation itself, comparing LLM scales, augmentation configuration under token constraints, finding that augmentation quality scales with contextual richness, albeit at a proportionally higher cost. Experiments with SASRec demonstrate the effectiveness of the proposed approach, achieving over 100% and 60% improvement in group-specific NDCG@20 for extreme and regular long-tail items, respectively, on our industrial dataset, and 40%, 25% for MovieLens-32M, while increasing their coverage without significant overall degradation. We further validate the framework on a deployed production ranker with proprietary data. Under production cost constraint, label insertion at 10% augmentation, long-tail group NDCG@20 improves by 24%, while overall NDCG@20 decreases by only 1%.

1 Introduction

Collaborative filtering models learn latent item representations from user-item interactions, but perform poorly when items have inherently sparse interaction data (He et al., 2017; Wang et al., 2019). Popularity bias creates a feedback loop that systematically suppresses exposure for items with insufficient co-occurrence signals for both of long-tail and cold-start content (Mansoury et al., 2020; Zhu et al., 2021). Within our industrial dataset (PRD) and ML-32M (Smucker and Chamani, 2025), the top 15% of items account for 95% and 98% of all interactions, respectively. Baseline SASRec (Kang and McAuley, 2018) exhibits 92-95% drop from head to tail (Fig 1), with cold-start items receiving zero exposure.

Traditional approaches such as self-supervised learning (Zhou et al., 2020), contrastive sequence augmentation (Xie et al., 2022), and adaptive negative sampling (Chen et al., 2022) address the interaction sparsity while operating within the existing interaction distribution without introducing new exposure for underrepresented items. More recently, LLM-based approaches have been introduced within the recommendation paradigm, generating pairwise sequences (Wei et al., 2024), augmenting interaction graphs (Zhang et al., 2025; Xu et al., 2025), metadata-based item enrichment (Chen et al., 2025), and serving LLMs as recommenders for cold-start items (Sanner et al., 2023). Kim et al. (2024) align LLM token spaces with item embeddings, while Huang et al. (2025) simulate user item interactions from filtered candidate sets, and Wang et al. (2024) cluster LLM recommendations. Other optimizations include training an RL policy for efficient user selection in LLM augmentation (Subbaraman et al., 2025) and post-training LLMs for cold-start reasoning (Li et al., 2025). In-context learning has also been applied to build synthetic interaction matrices for cold-start users (Liang et al., 2025; Che et al., 2024).

Recent work has focused more on addressing long-tail recommendations, improving collaborative signals via self-distillation with semantic embeddings (Liu et al., 2024), and retrieval-augmented LLMs (Wu et al., 2024), both enhancing existing representations. Towards generating synthetic interactions, TADA (Dang et al., 2026) introduces tail-aware augmentation with heuristic sequence updates. Synthetic interactions generated with LLM have proven to serve as a reliable augmentation signal on a scale (Glisovic et al., 2026; Wang et al., 2024; Sanner et al., 2023), reinforcing that LLMs possess sufficient reasoning to act as effective data augmenters. However, these approaches either treat the long-tail as a monolith or do not characterize the tradeoff between overall and group-specific ranking quality under augmentation. Wang et al. (2025) underscores the importance of exploring cold-start items, but does not differentiate between popularity tiers. We build on this by

introducing a percentile-based grouping strategy over the interaction distribution, augmenting user histories with LLM-generated interactions drawn from group-specific candidate pools, and analyzing resulting trade-off between overall and group-specific ranking quality.

Our contributions are summarized as follows:

- We propose a group-wise targeted augmentation strategy that classifies items by interaction frequency to tailor the augmenting approach for each group, adapting to the dataset's popularity distribution.
- We characterize the trade-off between group-specific and overall metrics by estimating the Pareto frontier and examining their correlated regions, demonstrating that each group induces a distinct trade-off curve determined by its representation in the historical data. This highlights that a single strategy does not generalize across groups and that one-size-fits-all approaches are suboptimal.
- We demonstrate that individually optimal per-group strategies do not compose into a globally optimal solution, confirming cross-group augmentation as a multi-objective optimization problem requiring joint calibration.
- We perform a comprehensive ablation study over candidate selection, LLM scale, insertion position, augmentation percentage, and configurations, introducing a stepwise experimentation framework that narrows the design space without exhaustive search.
- We validate the industrial applicability of our framework by extending the evaluation beyond SASRec to a deployed production ranker using model real-world proprietary data, demonstrating the transferability of targeted augmentation across model and data settings.

2 Methodology

2.1 Dataset and Grouping Strategies

We evaluate MovieLens-32M (ML-32M)(Smucker and Chamani, 2025), and our **P**roprietary **R**eal-world **D**ataset (PRD) comprising user watch history (TV shows, movies, sports, etc.) from a large-scale entertainment platform. We partition the item catalog in each dataset $\mathcal{I}$ into item-interaction frequency groups at the 75^{th} and 95^{th} percentile computed in log-space over each dataset's interaction distribution. We select these boundaries based on the baseline SASRec's performance across the candidate splits, as this partition produces groups with measurably distinct NDCG@20: G_3 produce closer to the head, G_2 has close to a 95% drop, while G_0/G_1 score near-zero performance (Fig 1).

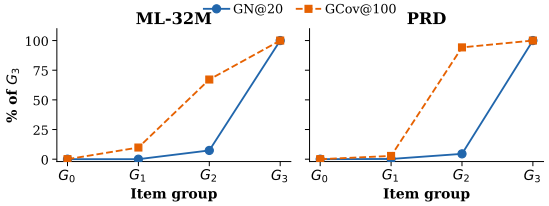


Figure 1: Original Group-wise NDCG@20 (GN@20) by item popularity group on ML-32M (left) and PRD (right)

Group	Percentile	ML-32M		PRD	
		Items	Range	Items	Range
G_0	N/A	71	0	2,755	0
G_1	0-75^{th}	62,645	[1-23]	225,806	[1-30]
G_2	75^{th}-95^{th}	16,414	[23 - 731]	60,930	[30-302]
G_3	95^{th}+	5,124	731+	16,671	302+

Table 1: Group distribution of items (full labeled dataset) and training-data interaction ranges across datasets

Unlike fixed threshold approaches that define grouping by arbitrary interaction thresholds, these percentile-based groupings adapt to each dataset's interaction distribution, as the same split yields different absolute thresholds on ML-32M and PRD, reflecting their distinct popularity curves, where G_1 (0-75^{th}) and G_2 (75^{th} - 95^{th}) capture extreme and regular long-tail items respectively, while G_3 is the popular tier and G_0 are cold-start items.

2.2 Augmentation Pipeline

`Llama-3.3-70B-Instruct` (Grattafiori et al., 2024) serves as the default synthetic interaction generator. For each target group $G_k \in \{G_0, G_1, G_2\}$, the pipeline randomly samples a proportion (p) of users and augments each with a single synthetic item ($i = 1$) via LLM, selected from among $c = 10$ candidates drawn from G_k. The augmentation per user is motivated by previous literature (Wang et al., 2024; Subbaraman et al., 2025), where a single synthetic interaction per user minimizes the risk of shifting the users' preference distribution while still providing a meaningful signal to the base recommender. Candidate selection is performed via cosine similarity between Linq-Embed-Mistral (Junseong Kim et al., 2024) item embeddings and mean-pooled users' representation on **PRD**, determined by the encoder ablation

in Table 2. On *ML-32M*, where item metadata is insufficient for meaningful embedding construction, random sampling is used to select candidates for augmentation. The LLM selects from a closed can-

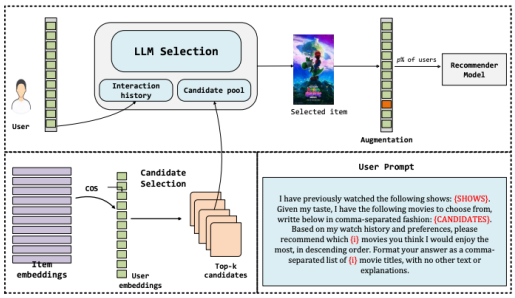


Figure 2: Overview of the data augmentation pipeline.

didate pool represented in the prompt (Appendix A), eliminating out-of-catalog selections. The selected item is inserted at S_{n-2}, in the user's interaction sequence $S_u = (s_1, s_2, .., s_n)$ immediately preceding the validation (s_{n-1}) and test (s_n) items. SASRec is retrained from scratch on the augmented interaction matrix. All metrics are averaged over three independent runs. The prompt templates and implementation details are provided in Appendix A. We formulate evaluation as a constrained optimization, group-specific NDCG@20 (GN@20) and group coverage GCov@100 that measures each group's share of top-100 recommendations per user, are the objectives to maximize, subject to overall NDCG@20 (N@20) degradation remaining within an acceptable threshold.

2.3 Experiment Designs

Given the vast search space, optimizing all variables concurrently results in a combinatorial explosion. Therefore, following prior work (Subbaraman et al., 2025; Wang et al., 2024; Liang et al., 2025), we fix selected variables, such as the quantity and insertion position of the augmented data, and prioritize the most impactful axes of variation. To systematically navigate the remaining search space, we adopt a coordinate-descent-inspired strategy that sequentially fixes hyperparameters to isolate and optimize individual variables. This stepwise optimization pipeline provides a practical and reproducible procedure across datasets and models. The following studies isolate the effects of candidate selection, augmentation proportion, cross-group allocation, and LLM configuration, followed by a production-model validation that examines whether the framework transfers beyond the benchmark SASRec setting.

Candidate Selection Optimization: We compare embedding encoders from the MTEB leaderboard (Muennighoff et al., 2022; Enevoldsen et al., 2025): `Linq-Embed-Mistral` (Mistral-7B) (Junseong Kim et al., 2024), `qwen3-4b` (Yang et al., 2025), `gte-large-en` (Li et al., 2023; Zhang et al., 2024) on G_2 at $p = 5\%$, selecting the encoder that maximizes group-specific NDCG@20. We validate the best encoder against random sampling across several augmentation proportions and quantify the LLM's marginal contribution by comparing paired settings with and without LLM reranking (Table 2 (b)). On ML-32M, we randomly sample the candidates due to lack of sufficient item metadata.

Pareto-front analysis: Given the best candidate selection strategy from **(1)**, we vary the augmentation proportion p to identify both the group-specific Pareto front, finding the trade-off boundary, and a correlated region where targeted and guardrail metrics improve simultaneously.

Cross Augmentation Optimization: Depending on the industry's objectives, optimizing performance across all groups may be necessary rather than focusing on a single specific group. we augment G_0, G_1, and G_2 simultaneously under varying augmentation splits, inspired by the previous step, to evaluate the optimal cross-group augmentation strategy.

LLM Design Space Optimization: We evaluate whether targeted solutions can be further improved using LLM optimization techniques under a cost constraint. We fix G_2 at $p = 5\%$, and perform ablation studies over LLM scale (8B, 70B, 405B), insertion position ($S_{n/2}$ vs S_{n-2}), user-profiles, and LLM-as-a-judge (Fabbri et al., 2025) to isolate each component's individual contribution towards overall and group-wise metrics. We note that this specific setup serves as an illustrative benchmark to demonstrate the optimization framework, which can be similarly applied to other groups or extended across all groups.

Case Study: Production Model and Offline evaluation: Building on augmenter ablations, we extend the G_2 evaluation on our production ranker model (PRM) with offline performance metrics. SASRec follows an autoregressive objective, deriving labels at every position in the sequence. PRM treats all of the user history (S_0 to S_{n-1}), as a single input with one label at S_n . Therefore, we introduce another design choice on insertion position, whether to augment the history (S_{n-1}) or the label

embeddings	N@20	GN@20
mistral-7B	+6.36	+15.97
Qwen-3B	+4.79	+6.42
GTE-en	-6.47	+27.49

p	random	mistral-7b
1%	-0.79	+ 8.14
5%	+24.87	+48.28
10%	+31.15	+62.42

Table 2: Candidate selection optimization (PRD) (G_2): (left) encoder-only (right) $\Delta GN@20(\%)$: best encoder vs. random.

Candidate Selection	Augmenter	ΔN@20(%)	ΔGN@20(%)
Random	-	-3.44	-13.04
Random	Llama-70B	-4.03	+24.87
mistral-7B	-	+5.59	+15.97
mistral-7B	Llama-70B	-2.09	+48.28

Table 3: Comparing two baselines vs LLM reranking across candidate selections; G_2: $p = 5\%$.

(S_n). We evaluate both insertion strategies using Llama-8B (Touvron et al., 2023) ($i = 1$) across p $\in \{1\%, 3\%, 5\%, 10\%, 15\%\}$. As p is model and data-specific, the production and benchmark grids identify an optimized operating point within each setting rather than support a direct comparison with Table 6. The offline performance of PRM is evaluated by ranking the predicted scores of the model for the items observed in each user's impression set and comparing it with the engagement of the user on those items. This impression-based evaluation differs from Fig 3's that used SASRec offline evaluation strategy.

3 Results

1) Candidate Selection Optimization: Table 2 (left) compares different embedding-based models for G_2 augmentation in PRD dataset at $p = 5\%$. The encoder determines the quality of the candidate pool presented to the LLM, providing it with an informed selection space by surfacing the most relevant items with respect to each user. Mistral-7B achieves the strongest balance between the overall and group-specific performance and consistently improves against random selection at all p values (Table 2 (right)), confirming embedding-based selections improve across all metrics. Paired with Llama-70B (Table-3), we further improve GN@20 by three times. Random selection degrades both metrics, as sampling from a larger vocabulary produces no semantic relevance to the user. *Mistral-7B* with Llama-70B is used as the default augmentation configuration for all downstream experiments. Embedding-based selection alone yields +5.59% N@20 and +15.97% GN@20 (Table 2), validating that cosine similarity already surfaces relevant candidates. Appending LLM to the pipeline boosts GN@20 to +48.28%. In addition, the first two rows in Table 3 shows that LLM augmenter is better than random in selecting augmented data.

2) Pareto-front analysis: As shown in Figure 3-a, each group traces a distinct tradeoff curve and optimal point governed by both deployment targeted metrics objectives and acceptable degradation in overall performance. We define a guardrail constraint of $\leq 5\%$ *N@20* degradation for identifying viable configurations. Within PRD, both G_2 and G_1 show positive correlated and tradeoff regions. In G_2, at $p = 1\%$, overall and group NDCG metrics improve by +3.8% and +8.14% respectively, targeted metrics continue to rise (GN. of +62% and GCov. of +5% in $p = 15\%$) while N@20 remains within -2% of the baseline throughout, suggesting that G_2 is the most tolerant group to augmentation. Augmenting items that already posses partial learned representations provides SASRec with additional informative signals, allowing the model to learn their representations better, without introducing a sufficient distributional shift that degrades the overall performance. G_1 items are extremely sparse in the training data, resulting in under-trained item embeddings sensitive to small volumes of synthetic augmentations. At lower p, synthetic interactions can be insufficient to overcome the noise, and their effect becomes volatile and smaller. At higher volumes ($p \geq 10\%$), G_1 shows consistent group-wise improvement, at the cost of sacrificing the overall performance.

The lack of any N@20 improvements for ML-32M in G_1 and G_2 plots can be attributed to the candidate selection strategy. For G_1, ML-32M evaluates at GN@100 due to limited test users (660) as GN@20 yields near-zero across all configurations. Random candidate selection can introduce noise that prevents joint metrics optimization. Due to the varying distribution of long-tail groups across datasets, the gain in targeted metrics may differ; however, consistent trends are observed across them. For example, given a fixed p, G_1 shows a larger improvement in the metric than G2 across both datasets, as it starts from a substantially lower baseline. We select $p = 3\%$ as $G_1's$ operating point for further cross-group augmentation experiments as it falls in the correlated region while having the lowest augmentation budget to minimize the overall augmentation proportions (Dang et al., 2026).

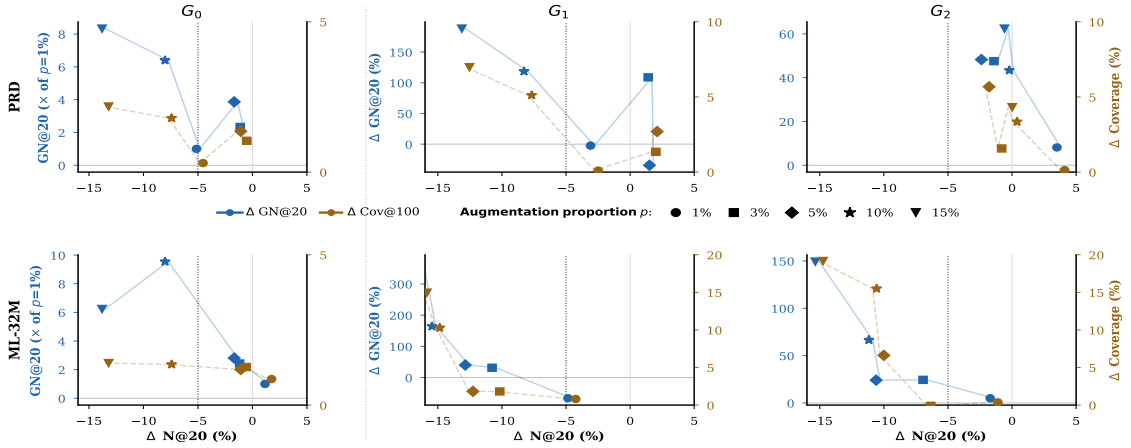


Figure 3: N@20-GN@20 and N@20-GCov@100 tradeoffs across different (p). The black dotted line indicates the predefined constraint for the guardrail metric. With a zero baseline for G_k, improvements are reported relative to 1% (p). All reported numbers are averaged across three runs and are statistically significant.

$[G_0(p), G_1(p), G_2(p)]$	N@20	G0=N@20	G1-N@20	G2-N@20
$[5, 3, 15]$	-24.49%	$2.96x$	+53.95%	**+45.67%**
$[5, 5, 5]$	-21.38%	$5.74x$	+11.31%	+19.86%
$[1.66, 1.66, 1.66]$	**-1.06%**	$2.87x$	-1.46%	+20.59%
$[1.92, 1.16, 1.92]$	-11.04%	$2x$	-7.69%	+38.41%
$[3.84, 2.32, 3.84]$	-14.95%	$4.17x$	-41.06%	+17.95%
$[5, 3, 5]$	-13.8%	$5.57x$	+32.24%	+19.24%
$[5.77, 3.46, 5.77]$	-20.6%	$6.61x$	**+71.52%**	13.44%

Table 4: PRD cross augmentation opt. via p distribution.

Augmenter	ΔN@20(%)	ΔGN@20(%)	token (x)
Llama-70B-default (baseline)	-2.09	+48.28	1.0x
Llama-8B	-0.15	+43.45	1.0x
Llama-405B	+3.11	+48.55	1.0x
Llama-70B (early: $S_{n/2}$)	+0.06	+31.67	$1.0x$
LLM-as-a-judge (Fabbri et al., 2025)	**+6.36**	+34.93	$2.06x$
Llama-70B + user profiles	-0.16	+34.09	$1.57x$
Llama-70B + metadata	+3.78	**+60.39**	$5.12x$

Table 5: PRD augmenter ablation performance (%)

Cold-start items retain their random initialization throughout training, receiving no gradient updates from real interactions, and augmentation provides the only source of learning signal for these embeddings. At $p \leq 5\%$, the augmentation is small enough for the model to improve both targeted metrics while maintaining the N@20 within the acceptable threshold. However, beyond this, the distribution shift steepens the tradeoff sharply. The optimal augmentation is $p = 5\%$, balancing a $4x$ GN@20 and $2x$ GCov@100 improvement against $\leq 5\%$ N@20 degradation. On ML-32M, the correlated regions can only be observed at G_0 for $p = 1\%$, owing to significantly smaller candidate pool for augmentation (71 vs 2.7k).

3) Cross Augmentation Optimization: We evaluate whether per-group augmentation strategies compose additively. Naively combining independent optimal proportions (23% total) degrades N@20 by -24.49%. This drop may stem from the total augmentation volume itself, their interaction effects, or both. We cap the overall augmentation proportion to $\leq 15\%$ as this represents the maximum evaluated in our per group analysis (Figure 3), to keep the guardrail metric decline, N@20 below the certain threshold and measure additive effects in joint optimization.

The non-uniform p allocation across groups and their respective shares of the total augmentation budget is motivated by patterns observed in single-group trade-off optimization. At 5% total cross-augmentation with equal splits, both G_0 and G_2 improve across 3rd and 4th rows, while non-uniform allocation yields a greater overall gain. However, G1 remains sensitive to the splits, improving at higher p that is consistent with our findings that G_1 requires a minimum augmentation volume to produce stable gains. No configuration simultaneously improves all groups without overall degradation. Applying group specific p inspired by previous results, last two rows, simultaneously improve all group specific metrics. However, to mitigate degradation in guardrail metrics, further calibration of the joint augmentation is required, framing it as a multi-objective optimization problem.

4) LLM Design Space Optimization We fix G_2 at 5% augmentation across all ablations for consistency and evaluate against several augmentation strategies as presented in Table 5. Insertion position significantly affects efficacy, with early-augmented diluting their influence as SASRec weighs heavily toward more recent items (Oh and Cho, 2024;

	S_n			S_{n-1}		
p	NDCG@20	GN@20	GCov@20	NDCG@20	GN@20	GCov@20
1%	-0.6%	+22.0%	+9.9%	-1.8%	+ 25.9%	-1.85%
3%	-4.2%	+24.0%	+4.9%	-1.8%	+20.9%	-1.2%
5%	-0.6%	+12.3%	+4.3%	-2.9%	+19.1%	-2.2%
10%	-1.0%	+24.6%	+7.7%	-0.4%	+34.7%	-2.8%
15%	-1.2%	+39.2%	+22.0%	-4.5%	+21.3%	+0.3%

Table 6: PRM offline metrics with synthetic items inserted as labels (S_n) or at end of the user history (S_{n-1})

Kang and McAuley, 2018). User-profile generation (Bang and Song, 2025) and llm-as-a-judge (Fabbri et al., 2025) adopt a least-to-most prompting paradigm (Zhou et al., 2022), in which intermediate structure are leveraged to iteratively refine the final candidate selection, thereby enhancing overall ranking performance. Augmenting the prompt with additional metadata achieves the highest GN@20, indicating that richer contextual information improves the quality of the resulting augmentations. While these augmentation configurations yield stronger gains, the optimal configuration is itself a tradeoff between guardrail metrics and cost/computational scalability. The base configuration (Mistral-7B + Llama-70B) offers the most favorable cost-quality ratio for production deployment, while metadata enrichment achieves the strongest group-specific improvement at significantly higher computational cost. This motivates task-specific fine-tuning of smaller LLMs, which may yield both stronger augmentation reasoning and lower inference cost compared to prompting a general purpose large language model (Li et al., 2025).

5) Production-Model Validation: We evaluate whether the augmentation optimization framework generalizes to a deployed production model on G_2, using impression-based retrieval metrics (section 2.3). The regular long-tail group was selected because it achieved the business objective of improving both diversity and performance within this group. Label insertion (S_n) has the strongest gains, GN@20 improves by 39.2%, while only -1.2% drop in N@20 1.2% and +22% long-tail coverage gain. GN@20 and GCov@20 both degrade upto $p = 5\%$ before following a consistent trend, as observed in SASRec. Insertion at S_{n-1} achieves higher GN@20 at $p = 10\%$ (+34.7%, -0.4% N@20) but coverage declines at all levels. Label insertion directly optimizes the item to predict a long-tail item as the next action, improving both ranking and coverage. History insertion enriches the input representation with long-tail context, enabling the model to learn a better representation of the user for long tail preferences. However, as the target remain unchanged, the recommendation output does not shift towards surfacing more long-tail items, leaving coverage unaffected. Although the production response differs from the benchmark trend, $p = 15\%$ provides a clear targeted improvement with limited impact on overall performance. This result demonstrates that the proposed framework can identify an effective long-tail augmentation strategy for a deployed production ranker. While $p = 15\%$ achieves higher GN@20 improvement, we select $p = 10\%$ (+24.6% GN@20, -1.0% N@20)as the production operating point, which entails 50% higher inference cost. Once the viable operating point is achieved, the augmented data can persist across multiple training cycles, provided the guardrail constraints are satisfied. We calibrate the augmentation cadence to optimize for the LLM inference costs within 10% of the training budget.

4 Conclusion

We presented a group-wise LLM augmentation framework for sequential recommendation to improve the performance on i item. We evaluated SASRec performance on ML-32M and a large proprietary dataset by separating items into cold-start (G0), extreme long-tail (G1), regular long-tail (G2), and popular tiers (G3). We also demonstrated how the proposed approach can be applied to optimize a specific business objective metric for a production model. Across candidate selection, augmentation ratio, cross-group splits, and LLM-side design choices, the results show that augmentation is group dependent rather than governed by a single strategy. The best single-group settings do not carry over directly to the joint case, where cross-group augmentation remains a constrained multi-objective problem. Embedding-based candidate selection is more reliable than random selection when semantic data is available. Underrepresented groups differ in how much augmentation percentage can help with group specific performance gain and the trade off with overall ranking metrics. The LLM ablations also show that augmentation quality depends not only on LLM-model scale but also on how informative the prompt is. Production-ranker results provide preliminary evidence that targeted augmentation can transfer to operational data, while the deterioration at larger budgets reinforces the need for model- and dataset-specific calibration. Next step is to run an AB test for the production model and test the performance gain of regular long tail group.

5 Limitations

We evaluate with SASRec (Kang and McAuley, 2018) as our sole base recommender with two datasets, and then extending its applicability to our production model and pipelines. While the framework is independent of any architectures, the trade-off curves and optimal augmentation proportions may vary. Additionally, the proprietary dataset cannot be released, however, we provide the complete pipeline configurations and grouping algorithms in Appendix A to replicate the framework on any dataset. While our offline evaluations demonstrate the significance of how the framework improves the group-specific metrics, whether these translate to improvement in other online metrics necessitate A/B testing in production.ML-32M metadata was not fully available and that limit us to test the embedding-based selection on it. Augmentation quality is dependent on the LLM's reasoning over available context, with the most gains in metadata-rich settings as observed in the perfromance gap between ML-32M and PRD, as well as Table 5. Cross-group augmentation experiments indicate that per-group optima are not additive, as simultaneous augmentation across groups introduce competing distributional shifts over the same interaction dataset.

References

Seunghwan Bang and Hwanjun Song. 2025. Llm-based user profile management for recommender system. *arXiv preprint arXiv:2502.14541*.

Shangkun Che, Minjia Mao, and Hongyan Liu. 2024. New community cold-start recommendation: A novel large language model-based method.

Lei Chen, Chen Gao, Xiaoyi Du, Hengliang Luo, Depeng Jin, Yong Li, and Meng Wang. 2025. Enhancing id-based recommendation with large language models. *ACM Trans. Inf. Syst.*, 43(5).

Yongjun Chen, Jia Li, Zhiwei Liu, Nitish Shirish Keskar, Huan Wang, Julian McAuley, and Caiming Xiong. 2022. Generating negative samples for sequential recommendation. *Preprint*, arXiv:2208.03645.

Yizhou Dang, Zhifu Wei, Minhan Huang, Lianbo Ma, Jianzhe Zhao, Guibing Guo, and Xingwei Wang. 2026. Tail-aware data augmentation for long-tail sequential recommendation. In *Proceedings of the ACM Web Conference 2026*, pages 5886–5897.

Kenneth Enevoldsen, Isaac Chung, Imene Kerboua, Márton Kardos, Ashwin Mathur, David Stap, Jay Gala, Wissam Siblini, Dominik Krzemiński, Genta Indra Winata, Saba Sturua, Saiteja Utpala, Mathieu Ciancone, Marion Schaeffer, Gabriel Sequeira, Diganta Misra, Shreeya Dhakal, Jonathan Rystrøm, Roman Solomatin, and 67 others. 2025. Mmteb: Massive multilingual text embedding benchmark. *arXiv preprint arXiv:2502.13595*.

Francesco Fabbri, Gustavo Penha, Edoardo D'Amico, Alice Wang, Marco De Nadai, Jackie Doremus, Paul Gigioli, Andreas Damianou, Oskar Stål, and Mounia Lalmas. 2025. Evaluating podcast recommendations with profile-aware llm-as-a-judge. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, pages 1181–1186.

Natalija Glisovic, Martin Tegner, and Danica Kragic. 2026. Data augmentation with llms for cold start recommendation in e-commerce. In *European Conference on Information Retrieval*, pages 94–99. Springer.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 173–182, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Feiran Huang, Yuanchen Bei, Zhenghang Yang, Junyi Jiang, Hao Chen, Qijie Shen, Senzhang Wang, Fakhri Karray, and Philip S Yu. 2025. Large language model simulator for cold-start recommendation. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, WSDM '25, page 261–270, New York, NY, USA. Association for Computing Machinery.

Jihoon Kwon Junseong Kim, Seolhwa Lee et al. 2024. Linq-embed-mistral:elevating text retrieval with improved gpt data through task-specific control and quality refinement. Linq AI Research Blog.

Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 197–206.

Sein Kim, Hongseok Kang, Seungyoon Choi, Donghyun Kim, Minchul Yang, and Chanyoung Park. 2024. Large language models meet collaborative filtering: An efficient all-round llm-based recommender system. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 1395–1406, New York, NY, USA. Association for Computing Machinery.

Shijun Li, Yu Wang, Jin Wang, Ying Li, Joydeep Ghosh, and Anne Cocos. 2025. Llm reasoning for cold-start item recommendation. *arXiv preprint arXiv:2511.18261*.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

Xurong Liang, Vu Nguyen, Vuong Le, Paul Albert, and Julien Monteil. 2025. In-context learning for addressing user cold-start in sequential movie recommenders. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, RecSys '25, page 979–982, New York, NY, USA. Association for Computing Machinery.

Qidong Liu, Xian Wu, Yejing Wang, Zijian Zhang, Feng Tian, Yefeng Zheng, and Xiangyu Zhao. 2024. Llm-esr: large language models enhancement for long-tailed sequential recommendation. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.

Masoud Mansoury, Himan Abdollahpouri, Mykola Pechenizkiy, Bamshad Mobasher, and Robin Burke. 2020. Feedback loop and bias amplification in recommender systems. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 2145–2148, New York, NY, USA. Association for Computing Machinery.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

Jeonglyul Oh and Sungzoon Cho. 2024. Measuring recency bias in sequential recommendation systems. *arXiv preprint arXiv:2409.09722*.

Scott Sanner, Krisztian Balog, Filip Radlinski, Ben Wedin, and Lucas Dixon. 2023. Large language models are competitive near cold-start recommenders for language- and item-based preferences. In *Proceedings of the 17th ACM Conference on Recommender Systems*, RecSys '23, page 890–896, New York, NY, USA. Association for Computing Machinery.

Mark D. Smucker and Houmaan Chamani. 2025. Extending movielens-32m to provide new evaluation objectives. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, page 3520–3529, New York, NY, USA. Association for Computing Machinery.

Nachiket Subbaraman, Jaskinder Sarai, Aniruddh Nath, Lichan Hong, Lukasz Heldt, Li Wei, and Zhe Zhao. 2025. Selecting user histories to generate llm users for cold-start item recommendation. *Preprint*, arXiv:2511.21989.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Dong Wang, Junyi Jiao, Arnab Bhadury, Yaping Zhang, Mingyan Gao, and Onkar Dalal. 2025. Item-centric exploration for cold start problem. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, RecSys '25, page 987–990, New York, NY, USA. Association for Computing Machinery.

Jianling Wang, Haokai Lu, Yifan Liu, He Ma, Yueqi Wang, Yang Gu, Shuzhou Zhang, Ningren Han, Shuchao Bi, Lexi Baugher, Ed H. Chi, and Minmin Chen. 2024. Llms for user interest exploration in large-scale recommendation systems. In *Proceedings of the 18th ACM Conference on Recommender Systems*, RecSys '24, page 872–877, New York, NY, USA. Association for Computing Machinery.

Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 165–174, New York, NY, USA. Association for Computing Machinery.

Wei Wei, Xubin Ren, Jiabin Tang, Qinyong Wang, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2024. Llmrec: Large language models with graph augmentation for recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, WSDM '24, page 806–815, New York, NY, USA. Association for Computing Machinery.

Junda Wu, Cheng-Chun Chang, Tong Yu, Zhankui He, Jianing Wang, Yupeng Hou, and Julian McAuley. 2024. Coral: Collaborative retrieval-augmented large language models improve long-tail recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 3391–3401, New York, NY, USA. Association for Computing Machinery.

Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive learning for sequential recommendation. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 1259–1273.

Lanling Xu, Junjie Zhang, Bingqian Li, Jinpeng Wang, Sheng Chen, Wayne Xin Zhao, and Ji-Rong Wen. 2025. Tapping the potential of large language models as recommender systems: A comprehensive framework and empirical analysis. *ACM Trans. Knowl. Discov. Data*, 19(5).

An Yang, Anfeng Li, Baosong Yang, et al. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Haobo Zhang, Qiannan Zhu, and Zhicheng Dou. 2025. Enhancing reranking for recommendation with LLMs through user preference retrieval. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 658–671, Abu Dhabi, UAE. Association for Computational Linguistics.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, et al. 2024. mgte: Generalized long-context text representation and reranking models for multilingual text retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1393–1412.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 1893–1902, New York, NY, USA. Association for Computing Machinery.

Ziwei Zhu, Yun He, Xing Zhao, Yin Zhang, Jianling Wang, and James Caverlee. 2021. Popularity-opportunity bias in collaborative filtering. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, WSDM '21, page 85–93, New York, NY, USA. Association for Computing Machinery.

A Appendix

A.1 Pipeline configuration

Table 7 shows the default configuration used across all experiments unless specified otherwise. All components are interchangeable, and our approach could be substituted with any base recommender, embedding encoder, or instruction-following LLM.

Component	Implementation
Base recommender	SASRec (Kang and McAuley, 2018)
LLM augmenter	Llama-3.3-70B-Instruct (Grattafiori et al., 2024)
Candidate encoder	Linq-Embed-Mistral (Junseong Kim et al., 2024)
Candidate selection	Cosine similarity (top-c)
Items per user (i)	1
Candidate pool size (c)	10
Insertion position	s_{n-2}

Table 7: Default pipeline configuration. All components are interchangeable.

A.2 Percentile-Based Grouping

Algorithm 1 describes the grouping strategy. The procedure is agnostic to any dataset and could be applied to any interaction dataset without manual threshold selection. However, the optimal percentile boundaries may vary across domains and recommender architectures, and we recommend validating the split against the baseline group-level performance to ensure distinct tiers (Fig. 1).

Algorithm 1 Percentile-Based Item Grouping

Require: Training set $\mathcal{D}_{train}$, test/val items $\mathcal{I}_{test} \cup \mathcal{I}_{val}$
Ensure: Item groups $\{G_0, G_1, G_2, G_3\}$
Count interactions per item: $n_i \leftarrow |\{(u,i) \in \mathcal{D}_{train}\}|$
Compute log-counts: $\ell_i \leftarrow \log(n_i + 1)$
Remove outliers: exclude items where $|\ell_i - \mu| > 3\sigma$
Compute P_{75}, P_{95} over remaining ℓ_i
$G_1 \leftarrow \{i : \ell_i \leq P_{75}\}$
$G_2 \leftarrow \{i : P_{75} < \ell_i \leq P_{95}\}$
$G_3 \leftarrow \{i : \ell_i > P_{95}\}$
$G_0 \leftarrow \{i \in (\mathcal{I}_{test} \cup \mathcal{I}_{val}) : n_i = 0\}$

A.3 Prompt Template

The following listwise prompt template is used for all LLM-based candidate generation experiments. The user's interaction history (`SHOWS`), and candidate items (`CANDIDATES`) are populated per user.

I have previously watched the following shows: `{SHOWS}`. *Given my taste, I have the following movies to choose from, written below in comma-separated fashion:* `{CANDIDATES}`. *Based on my watch history and preferences, please recommend which* `{i}` *movies you think I would enjoy the most, in descending order. Format your answer as a comma-separated list of* `{i}` *movie titles, with no other text or explanations.*

Figure 4: Listwise prompt template for LLM-based candidate selection.

A.4 Coverage Analysis

Table 8 lists the shift in group-wise coverage when each group is augmented at its optimal proportion (p^*) from the per-group analysis. The PRD coverage analysis for the selected Pareto-optimal point (Figure 3-b) shows augmentation redistributes exposure from G_3 to the targeted group, increasing item diversity in recommendation outputs. Augmenting G_1 in PRD and G_0 in ML-32M also increases G_3 coverage; this happens due to the synthetic interaction that increases both overall and group metrics. Under the $\leq 5\%$ N@20 threshold, ML-32M's G_1 and G_2, the optimal p^* is limited to $p = 1\%$, resulting in negligible coverage shifts.

	$G_k(p^*)$	ΔG_0	ΔG_1	ΔG_2	ΔG_3
PRD	G_0 (5%)	+1.35	-0.38	+0.40	-1.38
	G_1 (3%)	0.0	+1.35	-2.77	+1.43
	G_2 (5%)	0.0	+0.46	+5.68	-6.14
ML	G_0 (3%)	+1.26	-3.92	-2.78	+5.43
	G_1 (1%)	0.0	+0.81	-5.16	+4.35
	G_2 (1%)	0.0	-0.90	+0.36	+0.54

Table 8: Group Coverage distribution shift for the selected Pareto-optimal (p) per Group ($G_k(p*)$). Each row shows the coverage shift across all groups when augmenting the target group.

A.5 Data Preprocessing

For both datasets, we follow SASRec style evaluation (Kang and McAuley, 2018), with the penultimate for validation, last interaction for testing, and the rest for training. Interaction sequences are ordered chronologically and truncated or zero-padded to a maximum length of 150. Users with fewer than K interactions are excluded from both datasets.