Candidate Grounding Matters: LLM-Guided Negative Selection for Production Recommenders

Anonymous ACL submission

Abstract

Negative sampling requires deciding which items are credible negatives and which candidate should provide the training signal. We study these as *candidate grounding* and *negative selection* when fine-tuning a production recommender. Among impression, popularity-based, and full-catalog random sources, impression candidates provide the most consistent offline relevance gains and are validated in a production A/B test. Holding the candidate pool, model checkpoint, architecture, and training objective fixed, we compare two negative-selection strategies. Both production-model loss selection and LLM-guided selection operate on the same impression pool of shown but unselected items. The combined pipeline improves NDCG by 75.0% at $K = 1$ and 23.6% at $K = 10$, and Recall by 71.3% and 8.4% at the same cutoffs, over the deployed model. The LLM selector reduces the false-negative rate of subsequent-watch to 0.337× that of production selection, corresponding to a 66.3% reduction. Overall, the results show that effective negative sampling requires both behaviorally grounded candidates and reliable selection within that set of candidates.

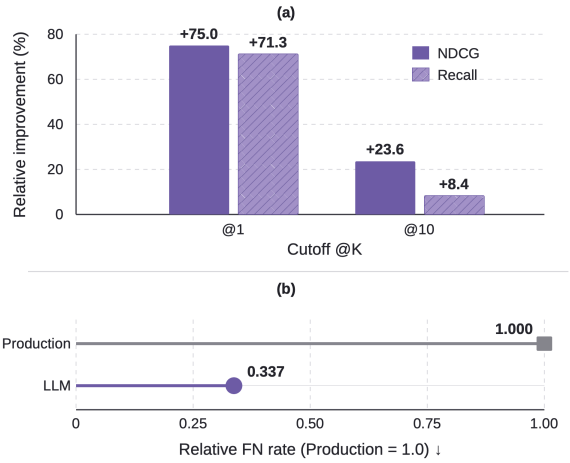


Figure 1: Overview of the main offline findings. (a) End-to-end NDCG and Recall improvements from exposure-grounded candidate construction followed by LLM-guided negative selection, relative to the deployed production model. We show representative shallow and deeper cutoffs at $K = 1$ and $K = 10$. (b) Subsequent-watch false-negative rate for LLM-guided selection relative to production-model loss selection within the same impression candidate pool (Production $= 1.0$). Lower is better.

1 Introduction

Negative sampling is a fundamental component of learning from implicit feedback, and the choice of sampling strategy can substantially affect the optimization of the recommender and the performance of the ranking.(Rendle et al., 2009; Petrov and Macdonald, 2023; Prakash et al., 2024; Ma et al., 2026). An observed interaction is a useful positive signal, but a missing interaction is harder to interpret. The user may dislike the item, may not have selected it yet, or may never have seen it (Saito et al., 2020). Random sampling is efficient, but it often produces easy negatives. Hard-negative mining provides a stronger training signal by selecting items that the model scores highly, but aggressive mining can also increase false-negative risk (Ding et al., 2020; Shi et al., 2023, 2025).

We treat negative sampling as two separate decisions. *Candidate grounding* determines which items are credible negative candidates by tying candidate eligibility to observable user behavior. *Negative selection* then chooses which candidate should provide the training signal. The distinction exposes two different failure modes. Catalog-wide sampling can produce negatives with weak behavioral evidence, while aggressive selection within a plausible candidate pool can choose items the user may actually prefer.

In this work, impression exposure provides the grounding signal. A non-clicked impression records an item that was shown to the user but was not selected, giving a stronger behavioral basis for

negative candidacy than an arbitrary unobserved item (Ding et al., 2019). We compare this exposure-based construction with popularity-based and full-catalog random candidates, then validate the selected exposure-based construction in a production A/B test. This online experiment exclusively evaluates the candidate-grounding stage.

Subsequently, we examine negative selection while holding candidate grounding fixed. For each eligible user, the LLM chooses one negative from the same impression candidate pool of up to 20 shown but unselected items using recent user history and candidate information. The comparison retains the production checkpoint, the recommender architecture, the training objective, and the candidate pool unchanged. This is to ensure that the downstream difference isolates the selector used during fine-tuning. Before scaling the LLM experiment, we also compare candidate LLM endpoints using semantic proximity, production-model hardness, rank statistics, and normalized inference cost.

Figure 1 summarizes both ranking quality and selector reliability. Figure 1 (a) shows the end-to-end gain from the two-stage pipeline. Relative to the deployed production model, NDCG improves by 75.0% at $K = 1$ and 23.6% at $K = 10$, while Recall improves by 71.3% and 8.4% at the same cutoffs. Figure 1 (b) holds the impression candidate pool fixed and reports the false-negative rate of the subsequent-watch relative to the selection of the loss of the production-model. The LLM selector reaches $0.337\times$ the reference rate, indicating a substantially lower false-negative risk than aggressive production-model loss selection. The remaining experiments separate candidate-grounding and selector effects and report complete diagnostic comparisons. Because LLM-guided selection incurs an offline inference cost, we also analyze sensitivity to the fraction of training users retained for LLM-guided fine-tuning.

Our contributions are as follows:

- We separate negative sampling into candidate grounding and negative selection. We compare impression-based, popularity-based, and full-catalog random sources, and validate the selected impression-based grounding strategy in a production A/B test.
- We isolate the effect of LLM-guided selection by holding the candidate pool, production checkpoint, architecture, and training objective fixed. This directly measures the value of the selector used during fine-tuning.
- We use subsequent user behavior to quantify false-negative risk. The results show that neither hard-negative selection nor false-negative avoidance alone is sufficient; effective negatives require both credible grounding and informative selection.
- We evaluate LLM selectors by quality, hardness, rank, and cost, and analyze selective LLM-guided fine-tuning to characterize practical quality–cost operating points.

2 Related Work

2.1 Negative Sampling and Candidate Grounding

Negative sampling is widely used to learn from implicit feedback, ranging from random sampling to dynamic hard-negative mining and knowledge-enhanced selection (Ma et al., 2026). Bayesian Personalized Ranking (BPR) formalizes pairwise learning between observed items and sampled unobserved items (Rendle et al., 2009). In sequential recommendation, both the sampling strategy and the number of negatives can change the optimization and ranking behavior. gSASRec, for instance, links conventional negative sampling to overconfidence in SASRec-style models and studies the effects of the loss function and the number of negatives (Petrov and Macdonald, 2023). Large-scale comparisons also report different accuracy and popularity-bias trade-offs for random, popularity-based, in-batch, and adaptive sampling (Prakash et al., 2024).

Hard-negative sampling focuses on training on difficult items, but the hardest candidates can also be false negatives. Prior work addresses this problem through sampling approaches that balance informativeness and reliability (Ding et al., 2020; Shi et al., 2023) and through objectives that reduce the effect of false negatives during hard-negative training (Shi et al., 2025). We study the same trade-off within a fixed candidate pool by comparing a production-model loss selector with an LLM selector and by checking whether selected negatives are later consumed.

The impression logs provide direct evidence of exposure. Without that information, a missing interaction may simply mean that the user never saw the item (Saito et al., 2020). Ding et al. (2019) use

exposed but non-interacted items as more informative negative candidates than arbitrary unobserved items. We use impression exposure as a candidate-grounding signal. The impression pool defines the set of behaviorally supported negative candidates, and the later experiments compare different selectors within that fixed pool.

2.2 LLM-Guided Negative Selection

Recent studies have explored LLMs to construct and select negatives in recommendation. `HNLMRec` generates semantic hard negatives and reports that behavioral grounding helps avoid noisy training samples (Zhao et al., 2025). `DTL-NS` combines LLM signals with collaborative information to identify false negatives and construct more reliable negatives (Wu et al., 2026). In an industrial retrieval setting, Ji et al. (2026) use LLM-based semantic clustering for hard-negative sampling at scale.

We use the LLM solely as a selector after candidate grounding. It receives the same fixed impression candidate pool as the non-LLM baseline and chooses one negative from that pool. This design separates candidate eligibility from selector behavior and does not require any change to the recommender architecture.

Selective use of LLMs has also been explored as a way to trade model quality for inference cost. `RouteLLM`, for instance, learns which requests should be sent to a stronger model (Ong et al., 2025). We include a smaller offline sensitivity analysis over the fraction of training users retained for LLM-guided fine-tuning, but treat it as an efficiency analysis rather than the main negative-selection method.

3 Experimental Setup

3.1 Evaluation Protocol

We evaluate a SASRec-like sequential recommender (Kang and McAuley, 2018). All experiments are initialized from the same trained production checkpoint, and they fine-tune that checkpoint under a different negative-sampling configuration. Prior work has shown that pre-trained sequential recommenders can be adapted to smaller task-specific domains through fine-tuning (Zivic et al., 2024). We study how candidate grounding and negative selection affect that fine-tuning stage. The offline dataset and preprocessing pipeline remain fixed. We choose the loss function and number of negative samples in the preliminary study reported in Appendix A, then keep both fixed for the remaining experiments. The recommender architecture is also unchanged. The frozen production checkpoint provides the candidate scores used to measure hardness and to define the user-ranking score used in the secondary LLM-share analysis.

We first compare the following non-LLM candidate sources:

- *Impression:* up to 20 non-clicked impression items for each eligible user, with 14.2 candidates per user on average. This is the exposure-grounded candidate source.
- *Popular:* 20 items randomly sampled from the top-1,000 popular items after excluding previously consumed items. This source is catalog-based rather than exposure-based.
- *Random:* one eligible random item sampled from the catalog. This source is also catalog-based.

For the *Impression* and *Popular* pools, we compute the loss for each candidate and choose the highest-loss item. The impression source gives the best overall balance between relevance and coverage, so we use that pool in the later LLM experiments. We refer to this fixed set as the *impression candidate pool*. In our experiments, impression exposure operationalizes candidate grounding, while hardness and selector behavior are measured within the resulting pool. Every configuration fine-tunes the same production checkpoint. We use *production-model loss selector* for the non-LLM selector that chooses the candidate with the highest loss under the frozen production model. We use *LLM selector* for the LLM that chooses one negative from the same impression candidate pool of up to 20 candidates using recent user history and candidate information. The selector is prompted to choose exactly one informative negative that is relevant enough to be non-trivial but unlikely to reflect the user's true preference. It uses title and genre/theme information, favors surface-level overlap with the user's history while considering deeper differences in tone, audience, or format, and avoids both trivial mismatches and candidates that may be false negatives. The full prompt is provided in Appendix B.

3.2 LLM Share Sensitivity

For each user u, let i_u^+ denote the observed positive item and $\mathcal{C}_u$ the impression candidate pool. Using

the frozen production recommender score $s(u, i)$, we define the positive–hardest-negative gap as

$$g_u = s(u, i_u^+) - \max_{c \in \mathcal{C}_u} s(u, c). \quad (1)$$

A smaller gap means that the production model scores the hardest candidate closer to the positive item, or above it. For the secondary efficiency analysis, we rank fine-tuning users by this gap and select a fraction for LLM-guided negative selection. At $p\%$ LLM share, only the selected $p\%$ of training users are used for fine-tuning. The selected user sets are nested across shares.

We first evaluated low and moderate LLM shares of 10%, 30%, and 50%. These settings did not preserve the core relevance metrics well enough to provide a useful operating point. We therefore focus the reported efficiency analysis on the high-coverage regime with 70%, 80%, 90%, and 100% LLM share. The score gap is used only to rank users when constructing these training subsets. The results are non-monotonic across LLM shares, so we treat this analysis as an empirical comparison of operating points rather than assuming that more LLM-selected training data always improves recommendation quality.

3.3 Production Validation of Candidate Grounding

We evaluate the selected exposure-grounded impression construction in a production A/B test before running the LLM selector experiments. The online test validates the candidate-grounding stage exclusively. LLM-guided negative selection is evaluated offline. We report relative treatment effects for the primary success guardrail and representative business and interaction metrics. The experiment passed the sample-ratio-mismatch check.

4 Results

4.1 Candidate Grounding

Table 1 compares candidate-source strategies that differ in behavioral grounding. The exposure-grounded impression source gives the most consistent relevance gains. NDCG improves by 25.0%, 19.0%, 13.0%, and 9.5% at $K = \{1, 3, 5, 10\}$, while Recall improves by 31.4%, 20.9%, 9.1%, and 7.2% at the same cutoffs. Random negatives perform well at the first position but are less consistent at deeper cutoffs. Popular negatives increase coverage substantially while reducing most ranking metrics. We therefore use impression exposure to ground the candidate pool for the production A/B test and all later selector experiments.

Metric	K	Imp.	Pop.	Rand.
NDCG	1	**+25.0%**	-25.0%	**+25.0%**
	3	**+19.0%**	-4.4%	+12.8%
	5	**+13.0%**	-14.5%	+4.0%
	10	**+9.5%**	-20.2%	+2.3%
Recall	1	**+31.4%**	-25.7%	+14.3%
	3	**+20.9%**	+4.0%	+2.0%
	5	**+9.1%**	-19.6%	-11.6%
	10	**+7.2%**	-27.2%	-7.7%
Coverage	1	+28.8%	**+44.0%**	-20.7%
	3	+24.3%	**+50.8%**	-15.9%
	5	+23.5%	**+55.0%**	-11.2%
	10	+19.5%	**+44.9%**	-7.0%

Table 1: Relative improvement (%) of candidate-source variants over the production configuration. The impression source provides exposure-grounded candidates, while Popular and Random are catalog-based. Bold values indicate the best result at each metric and cutoff.

Online Metric	Rel. Effect	Significance
Success (guardrail)	+0.16%	not significant
AVOD watch time	**+1.64%**	$p < 0.05$
Unique impressions	**+0.89%**	$p < 0.001$
Unique clicks	**+6.73%**	$p < 0.001$
Unique click rate	**+4.32%**	$p < 0.001$
Rail click	**+0.77%**	$p < 0.05$
Program app launch	**+0.99%**	$p < 0.05$

Table 2: Production A/B test of exposure-grounded impression candidate construction.

4.2 Production Validation of Candidate Grounding

Table 2 reports the production A/B test for the exposure-grounded impression candidate construction. The experiment passed the sample-ratio-mismatch check. Definitions of the online behavioral metrics are provided in Appendix C. LLM-guided negative selection is evaluated separately offline. The primary success guardrail remains statistically unchanged, while several engagement metrics improve significantly. Unique impressions increase by 0.89%, indicating a modest increase in the breadth of content exposed to users. In contrast, unique clicks increase by 6.73%, substantially more than the increase in exposure, and unique click rate improves by 4.32%. This pattern suggests that the gain is not driven simply by showing users more distinct content. Rather, a larger fraction of the exposed items leads to user interaction,

Model	Pos–Neg Cos. ↑	Mean Hard. ↑	Mean Model Rank ↓	Top-3 ↑	Rel. Cost ↓
GPT-5.4 Nano	0.5482	0.5306	**6.49**	**40.0%**	**1.0×**
GPT-5.4 Mini	0.5507	0.5114	7.11	33.0%	4.0×
GPT-6 Luna	0.5616	0.5190	6.81	27.4%	10.2×
GPT-5.4	**0.5693**	**0.5503**	6.84	35.4%	21.9×
GPT-6 Sol	0.5585	0.4921	7.40	24.7%	30.3×

Table 3: Comparison of LLM endpoints used as the LLM selector over the same impression candidate pool. Pos–Neg Cos. is computed from BGE-M3 (Chen et al., 2024) item embeddings. Hard. and Model Rank are measured under the frozen production model. Top-3 reports how often the selected negative falls among the three hardest candidates under the production-model loss ordering. Relative inference cost is normalized to the least expensive endpoint.

consistent with improved relevance of the ranked recommendations. The improvement also extends beyond item-level click behavior. Rail clicks increase by 0.77%, program app launches by 0.99%, and Advertising-based Video on Demand (AVOD) watch time by 1.64%. Together, these results suggest that the benefits of impression-based negative sampling extend beyond offline ranking metrics to measurable changes in user interaction and downstream engagement. This is consistent with the hypothesis that grounding negatives in observed exposure provides a more informative training signal than treating arbitrary unobserved items as negatives.

4.3 Selecting the LLM Endpoint

Before the downstream LLM selector experiments, we compare several endpoints using semantic similarity, hardness under the frozen production model, rank statistics, and relative inference cost. These measurements form a deployment-oriented screening protocol and are used only to choose the endpoint for the downstream selector experiments. GPT-5.4 Nano is competitive on hardness and rank-based measures and has the lowest normalized cost, so we use it in the remaining experiments. Table 3 reports the full endpoint comparison, and Appendix D defines the screening metrics.

4.4 LLM-Guided Selection within a Fixed Candidate Pool

With candidate grounding fixed, we fine-tune the same production checkpoint using either the production-model loss selector or the LLM selector. We use the impression-based baseline selected from the candidate-grounding comparison (Imp.) in Table 1, which provides the strongest overall trade-off among the evaluated candidate sources. This baseline uses the production-model loss selector over the impression candidate pool, while the LLM selects one negative from the same pool. Table 4 therefore measures the effect of replacing the production-model selector with LLM-guided selection while keeping the candidate grounding fixed.

Metric	@1	@3	@5	@10
NDCG	**+40.0%**	**+37.9%**	**+15.7%**	**+12.9%**
Recall	**+30.4%**	**+31.1%**	**+1.8%**	**+1.1%**
Coverage	-9.2%	-0.6%	**+3.7%**	**+5.4%**

Table 4: Relative improvement (%) of LLM-guided negative selection over the selected impression-based baseline.

Selector	Candidate Pool	Rel. FN Rate
LLM	Impression	0.337×
Production	Impression	1.000×
Random	Impression	0.346×
Random	Full catalog	**0.002**×

Table 5: False-negative (FN) diagnostic across selectors and candidate pools. A selected negative is counted as a false negative if the same user watches it later. Rates are normalized to the production-model loss selector, and lower is better. The full-catalog random row corresponds to the Random candidate source in Table 1.

Relative to this selected impression-based baseline, NDCG improves by 40.0%, 37.9%, 15.7%, and 12.9% at $K = \{1, 3, 5, 10\}$, and Recall improves at all four cutoffs. Coverage is lower at shallow cutoffs and becomes positive at $K = 5$ and $K = 10$.

4.5 Selector Reliability and False Negatives

We measure selector reliability by checking whether the user later watches a selected negative. Table 5 uses the production-model loss selector as the 1.000× reference. Within the same impression candidate pool, LLM-guided selection reduces the relative false-negative rate to 0.337×, while

Metric	K	70%	80%	90%
NDCG	1	+14.3%	+7.1%	+0.0%
	3	-0.9%	-3.3%	+2.2%
	5	+7.0%	+1.1%	+0.9%
	10	+4.7%	+1.9%	-0.6%
Recall	1	+13.3%	+10.0%	+5.0%
	3	-8.8%	-9.4%	+3.0%
	5	+10.2%	-1.5%	+1.6%
	10	+2.1%	-3.3%	-5.4%
Coverage	1	+1.6%	+2.7%	+0.0%
	3	+0.9%	+4.1%	+0.1%
	5	-0.5%	+3.6%	+0.0%
	10	+0.8%	+3.1%	+1.8%

Table 6: Relative change (%) when fine-tuning on high-coverage LLM-selected subsets compared with full LLM selection. At $p\%$ LLM share, only the selected $p\%$ of training users are used for fine-tuning. Full LLM selection is omitted because it is the 0.0% reference.

random selection reaches a similar 0.346×. This similarity shows that false-negative reduction itself is not unique to the LLM. The candidate pool also matters. Full-catalog random selection reduces the relative false-negative rate further to 0.002×, but Table 1 shows less consistent ranking gains at deeper cutoffs and substantially lower coverage for that source. The two analyses therefore capture different requirements. Candidate grounding determines whether negatives carry useful behavioral evidence, while the selector determines which grounded candidate supplies the training signal. Minimizing false negatives alone is not sufficient, while always selecting the highest-loss candidate under the production model produces substantially more false negatives. The ranking gains in Table 4 show that LLM-guided selection provides a stronger operating point within the same exposure-grounded pool.

4.6 Sensitivity to LLM Share

Table 6 reports results for high-coverage LLM-selected training subsets relative to full LLM selection. Preliminary experiments at 10%, 30%, and 50% LLM share substantially degraded core relevance metrics and did not provide a useful operating point. We therefore focus on LLM shares between 70% and 90%.

The high-coverage results reveal a clear trade-off rather than a monotonic relationship with LLM share. The 70% subset provides the strongest gains for several NDCG and Recall metrics, particularly at shallow and deeper cutoffs, while the 80% subset consistently provides the strongest coverage gains. The 90% subset is competitive at some intermediate cutoffs but does not consistently improve over either 70% or 80%. Because the subsets are nested and ranked by the production-model score gap, lower-share settings place more training weight on users whose hardest candidate is closer to the observed positive. This may concentrate learning on more informative examples, whereas adding users with larger score gaps may introduce easier examples with weaker training signals. At the same time, retaining more users can improve coverage, which explains why the best operating point depends on the target metric. Overall, these results suggest that selective LLM-guided fine-tuning can provide a useful quality–coverage–cost trade-off, but no single LLM share is uniformly best. The non-monotonic pattern also indicates that the production-model score gap is only an approximate heuristic for identifying where LLM-guided selection is most valuable. This sensitivity analysis can help inform practical operating-point selection by balancing downstream quality with LLM inference cost.

5 Conclusion

We study negative sampling through two separate decisions, 1) candidate grounding and 2) negative selection. Exposure-grounded impression candidates provide the most consistent offline balance between relevance and coverage, and the selected impression-based construction also produces gains on several interaction and business metrics in a production A/B test. Holding the impression candidate pool, the recommender checkpoint, the architecture, and the training objective fixed, LLM-guided selection further improves offline ranking quality and reduces the subsequent-watch false-negative rate to 0.337× that of production-model loss selection.

The results also show that false-negative reduction alone is not sufficient. Full-catalog random negatives achieve an even lower false-negative rate, but they show weaker ranking and coverage behavior than impression-grounded alternatives. Taken together, these findings suggest a practical design principle for production recommenders: first ground negative candidates in observable user behavior, then select training negatives within that candidate set while accounting for both ranking utility and false-negative risk.

Limitations

The study has four main limitations. First, we instantiate candidate grounding with impression exposure, so the results do not establish that other behavioral grounding signals would behave the same way. Impression candidates are also available only for users with eligible exposure history. Second, the production A/B test validates the candidate-grounding stage, while LLM-guided negative selection has not yet been evaluated in production. A production evaluation of the LLM selector is planned as a next step. Third, the endpoint comparison relies on semantic proximity and production-model hardness as screening measures rather than direct labels of negative quality. Finally, the LLM-share analysis uses a single production-model score gap as a subset-selection heuristic. Its non-monotonic results show that this signal does not reliably identify the users for whom LLM-guided fine-tuning is most valuable.

Ethical Considerations

This work investigates the use of large language models (LLMs) to choose a hard negative sample from a fixed impression candidate pool, given negative impression candidates with item metadata, and user's viewing history. All production systems internally go through rigorous privacy, legal and AI-specific legal review, in which we consider ethical use related to data privacy, fairness, and responsible use.

Data Privacy: The data provided to LLMs does not contain personally identifiable information (PII), instead, it contains the impression candidate with item metadata that's already utilized in a production model that sets the non-LLM baseline, and it contains the items that are present in user viewing history. Third-party hosted LLMs are used for processing this data. The association of the chosen hard negative to a user identifier is done in the backend processing of the pipelines. The data associated to teens and children profiles in TV platforms is not used in any part of our system.

Fairness and Bias: LLMs may exhibit biases arising from their training data, potentially influencing the chosen hard negative selected, which could positively or negatively affect diversity of recommendations. However, guardrails in our system binds the LLM-output to the fixed impression candidate pool, and drop any out-of-catalog responses, protecting against any unintended selections or choosing an out-of-bounds candidate.

References

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.

Jingtao Ding, Yuhan Quan, Xiangnan He, Yong Li, and Depeng Jin. 2019. Reinforced negative sampling for recommendation with exposure data. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 2230–2236.

Jingtao Ding, Yuhan Quan, Quanming Yao, Yong Li, and Depeng Jin. 2020. Simplify and robustify negative sampling for implicit collaborative filtering. In *Advances in Neural Information Processing Systems*, volume 33.

Ivan Ji, Liuyi Hu, Harrison (Zihao) Zhao, Lei Huang, Qunshu Zhang, Max Fan, and Aameek Singh. 2026. Real-time hard negative sampling via llm-based clustering for large-scale two-tower retrieval. *arXiv preprint arXiv:2607.00448*.

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

Haokai Ma, Ruobing Xie, Lei Meng, Fuli Feng, Xiaoyu Du, Xingwu Sun, Zhanhui Kang, and Xiangxu Meng. 2026. Negative sampling in recommendation: A survey and future directions. *ACM Transactions on Information Systems*, 44(4):1–44.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M. Waleed Kadous, and Ion Stoica. 2025. RouteLLM: Learning to route llms from preference data. In *The Thirteenth International Conference on Learning Representations*.

Aleksandr V. Petrov and Craig Macdonald. 2023. gsasrec: Reducing overconfidence in sequential recommendation trained with negative sampling. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 116–128.

Arushi Prakash, Dimitrios Bermperidis, and Srivas Chennu. 2024. Evaluating performance and bias of negative sampling in large-scale sequential recommendation models. *arXiv preprint arXiv:2410.17276*.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, pages 452–461.

Model	NDCG@3	NDCG@5	NDCG@10	Revenue
1_triplet	-6.5%	-8.9%	-10.8%	-23.7%
1_infonce	-0.5%	-2.7%	-4.8%	-26.0%
1_margin_softmax	-9.6%	-12.1%	-14.4%	-28.2%
1_hybrid	-1.7%	-4.0%	-6.1%	-23.0%
5_triplet	+7.3%	+5.3%	+3.0%	-8.1%
5_infonce	+8.3%	+6.4%	+4.1%	-14.8%
5_margin_softmax	+7.3%	+5.6%	+3.0%	-15.4%
5_hybrid	+8.1%	+6.4%	+4.0%	-14.7%
10_triplet	+6.7%	+5.1%	+3.4%	-1.6%
10_infonce	+8.4%	+6.8%	+5.1%	-9.0%
10_margin_softmax	+8.1%	+6.5%	+4.9%	-14.1%
10_hybrid	+6.5%	+5.2%	+3.8%	-12.6%
20_triplet	**+4.0%**	**+2.7%**	**+1.3%**	**+1.6%**
20_infonce	+2.1%	+2.0%	+0.9%	-10.1%
20_margin_softmax	+5.4%	+4.4%	+3.1%	-5.0%
20_hybrid	+4.7%	+3.8%	+2.3%	-7.7%

Table 7: Relative changes (%) from the production configuration. NDCG is reported at $K = \{3, 5, 10\}$. The selected configuration is shown in bold.

Yuta Saito, Suguru Yaginuma, Yuta Nishino, Hayato Sakata, and Kazuhide Nakata. 2020. Unbiased recommender learning from missing-not-at-random implicit feedback. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 501–509.

Kexin Shi, Jing Zhang, Linjiajie Fang, Wenjia Wang, and Bingyi Jing. 2025. Enhanced bayesian personalized ranking for robust hard negative sampling in recommender systems. *Statistics and Its Interface*. Accepted/In press.

Wentao Shi, Jiawei Chen, Fuli Feng, Jizhi Zhang, Junkang Wu, Chongming Gao, and Xiangnan He. 2023. On the theories behind hard negative sampling for recommendation. In *Proceedings of the ACM Web Conference 2023*, pages 812–822. ACM.

Jiayi Wu, Zhengyu Wu, Xunkai Li, Rong-Hua Li, and Guoren Wang. 2026. Dual-tree llm-enhanced negative sampling for implicit collaborative filtering. *arXiv preprint arXiv:2602.18249*.

Chu Zhao, Enneng Yang, Yuting Liu, Jianzhe Zhao, Guibing Guo, and Xingwei Wang. 2025. Can llm-driven hard negative sampling empower collaborative filtering? findings and potentials. *arXiv preprint arXiv:2504.04726*.

Pablo Zivic, Hernán Ceferino Vázquez, and Jorge Sánchez. 2024. Scaling sequential recommendation models with transformers. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1567–1577.

A Training Objective and Number of Negative Candidates

We compare different numbers of negative samples and training objectives using the same SASRec-like production model. Table 7 reports relative changes from the production configuration. We denote each configuration as `{number}_{loss}`, where `number` is the number of negative samples used per positive training instance and `loss` is the training objective. `InfoNCE` uses temperature $\tau = 0.1$, `Triplet` and `Margin Softmax` use margin $m = 0.2$, and Hybrid uses a triplet-component weight of 0.2.

We select `20_triplet` for the remaining experiments because it is the only tested configuration with positive changes across all reported ranking and revenue metrics. Relative to the production configuration, it improves NDCG@3 by 4.0%, NDCG@5 by 2.7%, NDCG@10 by 1.3%, and Revenue by 1.6%.

B LLM Negative-Selection Prompt

For each eligible user, the LLM receives recent user history and candidate titles with genre/theme tags from the exposure-grounded impression pool. We use the following selection instruction:

Negative-Selection Prompt

You are a recommendation training-data analyst. Your task is to select exactly one informative negative item for contrastive training. The selected item should be unlikely to match the user's true preferences, while still being sufficiently relevant that distinguishing it from positive items is non-trivial (i.e., a hard negative, not an obvious mismatch).
Each item is shown as: "Title" [genre/theme tags]. Use the tags to assess genre and theme overlap between history and candidates.

Guidelines:

- Prefer candidates that share surface-level genre or theme overlap with the history but diverge in deeper viewing signals (tone, audience, format).
- Avoid candidates that are clearly unrelated, as those are trivial negatives and provide no training signal.
- Avoid candidates that are too similar, as those risk being false negatives.

Title: <TITLES>
candidates: <CANDIDATES>
Return ONLY valid JSON. No additional text, no explanation.

```
{
  "negative_item":
     "<exact title from candidate items>"
}
```

C Online Metric Definitions

The offline evaluation uses NDCG, Recall, and recommendation coverage to measure ranking quality and recommendation breadth. The production A/B

test additionally reports behavioral metrics that capture user engagement with the recommendation experience.

AVOD watch time. Advertising-based Video on Demand (AVOD) watch time measures the amount of time users spend watching AVOD content. It reflects downstream viewing engagement after users interact with recommended content.

Unique impressions. Unique impressions measure the number of distinct content items exposed to a user within a session, counting repeated exposures of the same item only once. This metric captures the breadth of content presented during the recommendation experience.

Unique clicks. Unique clicks measure the number of distinct content items clicked within a session, counting repeated clicks on the same item only once. This metric captures the breadth of content with which the user actively engages.

Unique click rate. Unique click rate measures the fraction of distinct exposed items that are clicked within a session. Repeated impressions or clicks on the same item are counted only once, and only clicks on items that were also exposed in the session are included. The metric therefore captures how effectively distinct content exposures translate into user interaction.

Rail click. Rail click measures interactions with content items displayed within recommendation rails on browse or home surfaces. An increase indicates that users are more frequently engaging with the ranked content presented in these rails.

Program app launch. Program app launch measures cases in which a user selects a specific program and proceeds to launch the application providing that content. It represents a downstream action beyond the initial recommendation interaction and is therefore closer to content consumption intent than a browse-level click.

D LLM Selector Endpoint Screening

We compare LLM endpoints as alternative LLM selectors using the same user context and the same impression candidate pool. Because every candidate was exposed to the user but not selected, the pool already provides a behaviorally grounded set of negative candidates. Endpoint screening therefore focuses on which model provides the strongest selection signal from that grounded pool at an acceptable inference cost. We measure semantic proximity to the positive item, hardness under the production recommender, rank statistics, and relative inference cost.

Positive–negative cosine similarity. We encode the observed positive item and the LLM-selected negative with BGE-M3 (Chen et al., 2024) and compute

$$\cos(i_u^+, i_u^-) = \frac{\mathbf{e}_{i_u^+}^\top \mathbf{e}_{i_u^-}}{\|\mathbf{e}_{i_u^+}\| \|\mathbf{e}_{i_u^-}\|}. \tag{2}$$

Higher values indicate that the selected negative is semantically closer to the positive item. Within the impression-based candidate pool, this serves as a semantic proxy for selection difficulty. A selected item that is closer to the positive is more similar to content the user consumed and therefore provides a more challenging contrast for fine-tuning.

Production-model hardness. Let the frozen production recommender represent user u with embedding u_{emb} and score item i using item-specific model parameters W_i and b_i:

$$s(u, i) = W_i^\top u_{\text{emb}} + b_i. \tag{3}$$

Here, u_{emb} is the user representation produced by the frozen production model, while W_i and b_i denote the learned item-specific weight vector and bias used by its scoring layer.

For user u, let i_u^+ denote the observed positive item and c a candidate negative from the impression pool. We define the score gap between the positive and candidate as

$$\text{gap}(u, c) = s(u, i_u^+) - s(u, c). \tag{4}$$

A smaller gap indicates that the candidate receives a score closer to the positive item under the frozen production model, and therefore represents a harder negative. Conversely, a larger gap indicates an easier negative that is more clearly separated from the positive item.

Candidates are sorted by this gap in ascending order. Rank 1 is the hardest candidate within the impression pool because the production model scores it closest to, or above, the observed positive item. If the selected negative has rank r among K candidates, its normalized hardness percentile is

$$\text{HardnessPct} = 1 - \frac{r - 1}{K - 1}. \tag{5}$$

Rank 1 receives 1.0 and rank K receives 0.0. We set the value to 1.0 when $K = 1$. Uniform random

selection has an expected hardness percentile of approximately 0.5. Table 3 reports the mean hardness percentile across users.

Model rank and Top-3. The Mean Model Rank reports the average unnormalized rank r of the selected negative under the same production-model ordering. Lower values indicate harder candidates. Top-3 reports the fraction of users for whom the selected negative falls within the three hardest candidates:

$$\text{Top-3} = \frac{1}{N}\sum_{u=1}^{N}\mathbb{I}[r_u \le 3] \tag{6}$$

Relative inference cost. For endpoint m, we estimate the average LLM cost per user from its input and output token counts:

$$C_m = \frac{\bar{T}_m^{\text{in}}}{10^6}P_m^{\text{in}} + \frac{\bar{T}_m^{\text{out}}}{10^6}P_m^{\text{out}}, \tag{7}$$

where $\bar{T}_m^{\text{in}}$ and $\bar{T}_m^{\text{out}}$ are the average input and output token counts and P_m^{in} and P_m^{out} are the corresponding prices per million tokens. Relative cost is

$$\text{RelCost}_m = \frac{C_m}{\min_j C_j}. \tag{8}$$

The same ratio is obtained after scaling all endpoints to a fixed number of users, so the table reports only the normalized value. A relative cost of $1.0\times$ denotes the least expensive endpoint.

GPT-5.4 has the highest mean cosine similarity and mean hardness percentile. GPT-5.4 Nano has the best mean production-model rank and Top-3 rate among the screened endpoints. These results show that Nano frequently selects candidates that are difficult for the production model within the same behaviorally grounded pool while remaining close to the positive item in semantic space. The differences across endpoints are modest compared with the cost differences. Relative to Nano, Mini, Luna, GPT-5.4, and Sol cost approximately $4.0\times$, $10.2\times$, $21.9\times$, and $30.3\times$ more. We use GPT-5.4 Nano for the remaining experiments.