

GenRec: An LLM-Backed Recommendation Ranker at Netflix

arXiv: [v2] 21 Aug 2026

Introduction

- ❖ The recommendation system is central to the experiences of hundreds of millions of Netflix members, providing personalized recommendations across movies, series, games, live events, podcasts, and other content types.
- ❖ Netflix's existing production recommenders rely on extensive hand-crafted features spanning user, item, and interaction signals, together with higher-order feature interactions.

Introduction

- ❖ Although these systems have evolved over many years and are critical to the member experience, their complexity makes it increasingly difficult to support new business needs.
- ❖ Introducing a new content type or recommendation surface often requires substantial work in feature engineering, model architecture, infrastructure, and experimentation, slowing adaptation to growing business requirements.

Introduction

- ❖ LLMs offer broad world knowledge and expressive language understanding, enabling richer modeling of users, content, and context through natural-language representations of histories and contextual signals.
- ❖ Their flexibility also creates opportunities to adapt recommendations through prompts, reducing the need to redesign specialized models and pipelines for each new use case.
- ❖ However, off-the-shelf LLMs are insufficient for production recommendation: they may over-recommend popular content, hallucinate out-of-catalog titles, ignore business constraints, and provide limited personalization.

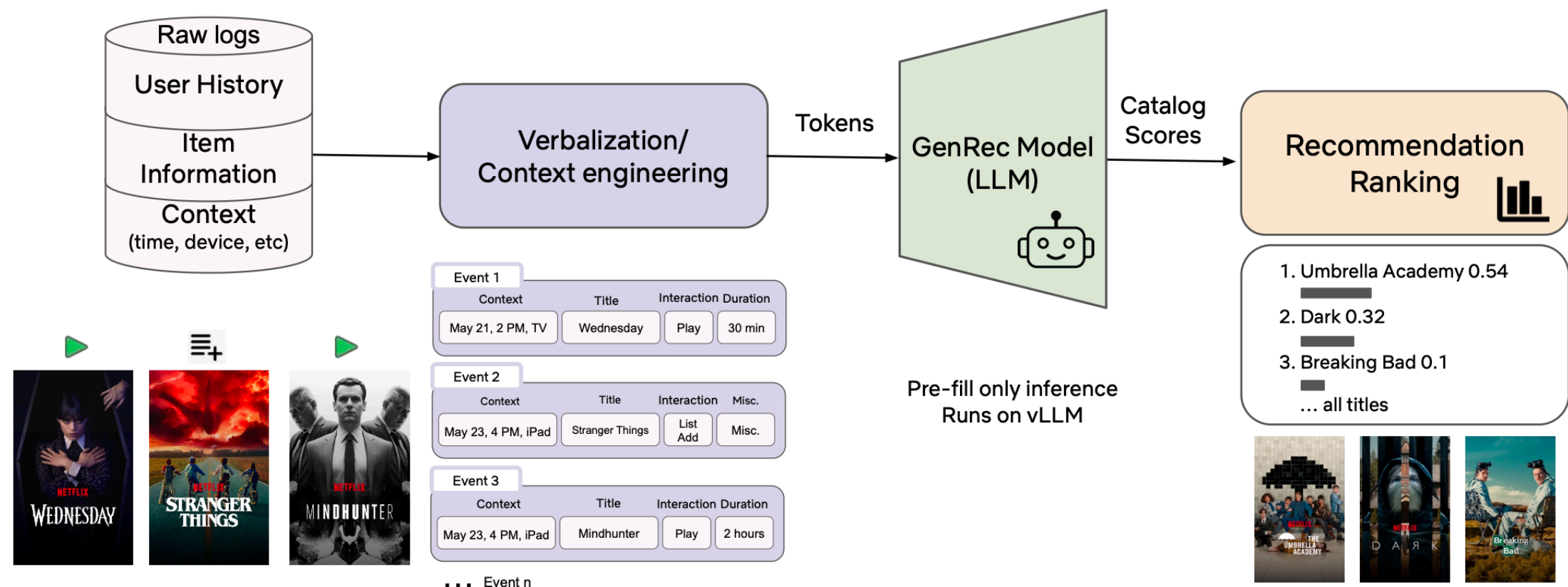
Introduction

- ❖ To address these issues, the authors introduce GenRec, an LLM-backed recommendation ranker focused on full-catalog ranking and intended as an initial step toward an LLM-native recommendation stack.
- ❖ GenRec follows a two-phase framework: Phase 1 adapts an open-source LLM to Netflix data, and Phase 2 further trains that foundation model for recommendation ranking.
- ❖ This article focuses on Phase 2, examining the transition from a traditional discriminative ranker with extensive engineered features to an LLM-backed ranker driven by verbalized user histories and context.

Note

- ❖ The paper provides an industrial system design and high-level evidence, rather than a fully reproducible experimental specification.
- ❖ It does not disclose exact dataset sizes, the base model identity, detailed train/test splits, selected loss coefficients, reward aggregation formulas, or precise definitions of the online metrics.

Method: Overview



- ❖ Figure 1 illustrates the GenRec inference pipeline. Raw logs of user interaction histories, item metadata, and contextual information are transformed through verbalization and context engineering into natural-language prompts. These prompts are tokenized and fed into GenRec, which runs on vLLM and outputs scores for the entire item catalog, inducing a personalized recommendation ranking.

Method: Problem Definition

- ❖ GenRec focuses on a full-catalog ranking problem, or top- K ranking if a candidate set is provided.
- ❖ **Notation:**
 - ❖ Let U denote the set of users,
 - ❖ C the catalog of items, such as movies, shows, games, live events, podcasts,
 - ❖ and X the space of contexts, such as device, surface, locale, and time of day.
- ❖ A recommendation request is characterized by a user $u \in U$, a context $\tau \in X$, and a time t .

Method: Problem Definition

- ❖ Let H denote the user's interaction history prior to t . Each item $i \in C$ is associated with metadata M_i , such as its title name, genre, synopsis, and release date.

- ❖ A recommender maps a request (u, τ, t, H) to a ranking π of the catalog C :

$$\pi : C \rightarrow \{1, \dots, |C|\},$$

- ❖ where $\pi(i)$ denotes the position assigned to item i , with rank 1 at the top of the list.
- ❖ The objective is to select a ranking that maximizes expected long-term member utility.

Method: Phases

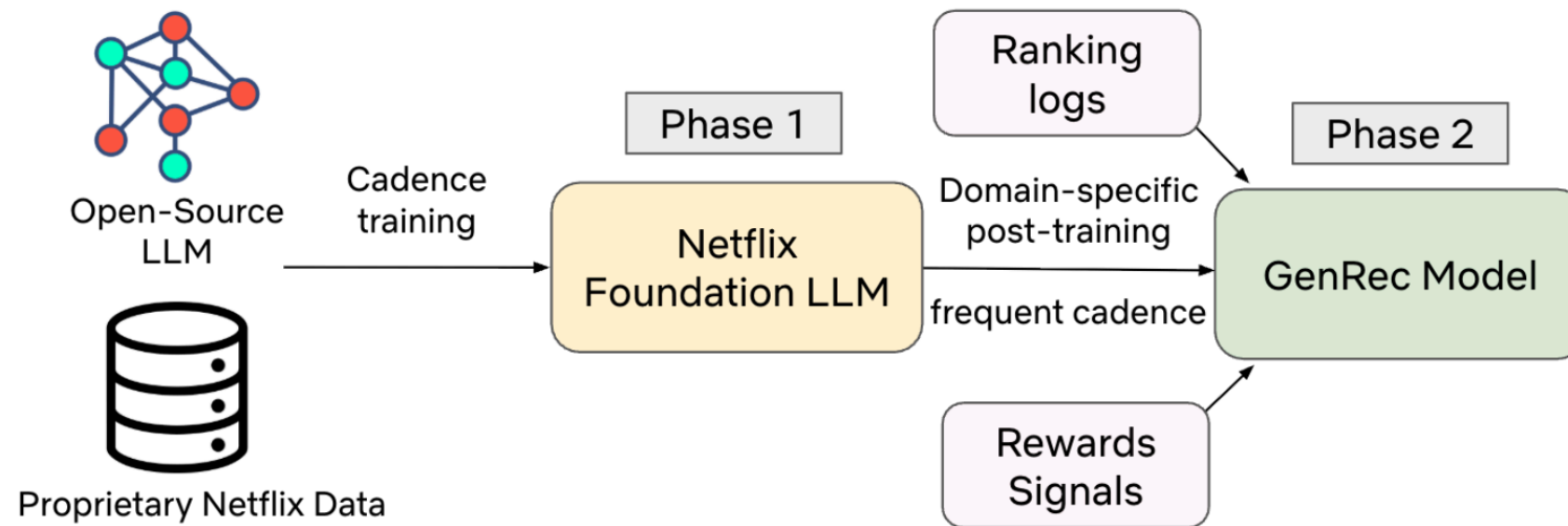


Figure 2: Two-phase training framework. Phase 1 trains a foundational LLM on Netflix data for user and content understanding, and Phase 2 post-trains on recommendation-ranking-specific data and objectives.

- ❖ To develop this ranker, GenRec follows a two-phase training framework.
- ❖ This paper focuses on Phase 2.

Method: Post-Training Data

- ❖ Historical interactions including plays, thumbs, and list additions are transformed into single-turn or multi-turn conversations between a user and a recommender.

- ❖ For a single-turn example:

$$(x, y) = (\text{user message, assistant message}) .$$

- ❖ x contains context, user profile and history, item information, and a task, such as predicting the next items the member will play.
- ❖ y represents actual member engagement—such as a play, duration, abandonment, or thumb rating—and provides ground-truth supervision.

Method: Context Engineering

- ❖ GenRec expresses histories and context as natural language or lightly structured text, allowing the LLM to learn item relationships, temporal dynamics, and evolving interests.
- ❖ Context engineering selects and compresses information to fit the token budget while controlling training and inference costs.
- ❖ High-signal engagements receive richer detail; noisy or low-signal events are omitted; repetitive behaviors such as binge-watching are compressed.
- ❖ Recent history receives greater detail, while older history is omitted or summarized.

Method: Post-Training Objective

- ❖ GenRec Phase-2 training combines mainly two main objectives:
 1. **Recommendation ranking objective:** The primary task is a catalog-aware ranking objective that trains the model to assign higher scores to items that correspond to high-value engagements.
 2. **Language modeling objective.** In addition, they include a language modeling objective over the verbalized inputs and outputs (such as predicted titles and other textual fields). This task helps maintain the backbone's general language understanding and text generation capabilities.

Method: Post-Training Objective

- ❖ The overall loss combines ranking, language modeling, and other objectives where applicable:

$$L = \alpha L_{\text{ranking}} + \beta L_{\text{language}} + \gamma L_{\text{miscellaneous}}$$

$$\alpha + \beta + \gamma = 1, \quad \alpha, \beta, \gamma \geq 0.$$

- ❖ The coefficients are are tunable hyperparameters which can be tuned through offline experiments.

Method: Model Architecture

- ❖ GenRec's backbone architecture closely follows the foundational LLM: **a decoder-only Transformer** trained with next-token-prediction style objectives.
- ❖ It is augmented with a **catalog-aware ranking head** that constrains recommendations to the Netflix in-catalog items and supports efficient scoring over large candidate sets.
- ❖ The scores are computed in the following three stages:

Method: Model Architecture

- ❖ **Verbalization:** A verbalizer V transforms interaction history H , item metadata $\{M_i\}_{i \in C}$, and context τ into a text sequence:

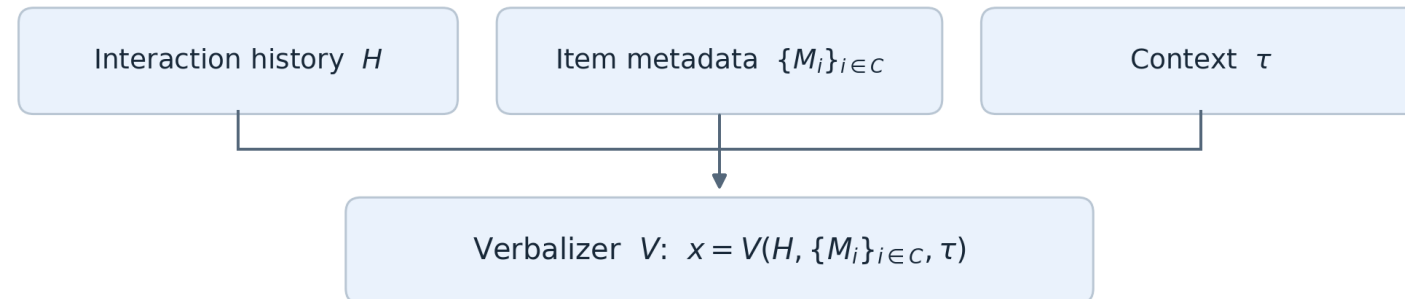
$$x = V(H, \{M_i\}_{i \in C}, \tau).$$

- ❖ **Pooled representation:** The LLM encodes x into a d -dimension representation h , taken as the hidden state at a pooling position, which summarizes the user preference and context.
- ❖ **Catalog-aware scoring:** A scoring head ϕ assigns each item i a score by combining h with a learned d -dimensional embedding e_i as the recommendation ranking task.

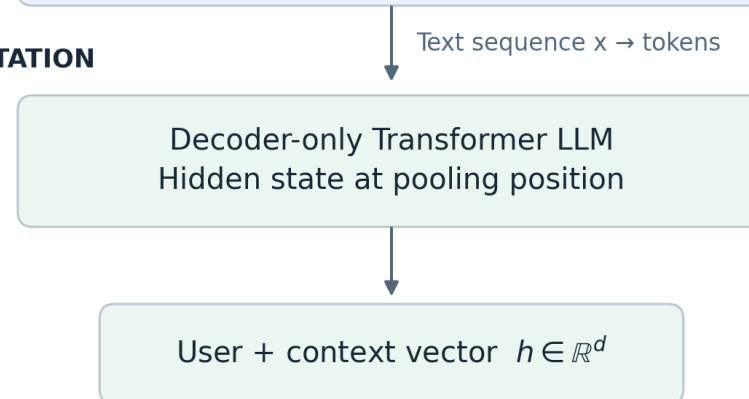
Method: Model Scoring

GenRec: model scoring

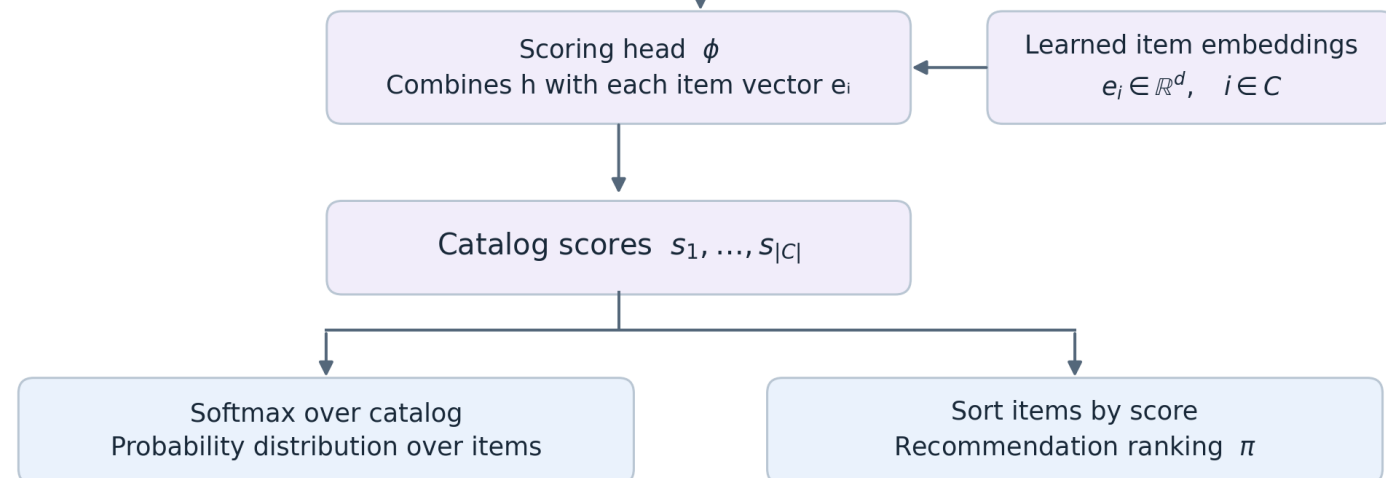
1 VERBALIZATION



2 POOLED REPRESENTATION



3 CATALOG-AWARE SCORING



Large catalogs: sampled softmax may be used

Jointly trained θ : LLM weights, scoring head, and item embeddings

Method: Reward Signals

- ❖ Beyond raw recommendation accuracy, GenRec must also satisfy two additional goals:
 1. **Respect business requirements**, such as content mixing logic (e.g., balance movies, shows, live, games, podcast, and other content types).
 2. **Maximize long-term member satisfaction**, rather than just optimizing short-term user engagements.

Method: Reward Signals

❖ To address, they use two broad categories of rewards:

1. **Long-term satisfaction proxies.** These rewards estimate how strongly a short-term engagement event correlates with long-term outcomes. Reward models assign higher values to engagements that encourage members to return to the service, explore across a wider portion of the catalog, or lead to sustained engagement over time.
2. **Behavior rebalancing across content and engagement types.** Additional rewards are used to **rebalance behavior across content types** (e.g., movies vs. games vs. live vs. pod-cast) and **launch stage** (e.g. pre-launch, newly launched, ever-green), in order to **meet business objectives** such as **exposure fairness across content types**.

Method: Reward Signals

- ❖ Outputs from multiple reward models are incorporated through a reward-weighted ranking loss.
- ❖ Each training example receives a scalar weight: high-value engagements receive larger weights, while low-value or undesirable behaviors are down-weighted.
- ❖ This approach provides a simple, stable, and cost-efficient alignment mechanism.
- ❖ Preliminary RL-style experiments, including GRPO, showed additional gains, but their higher training overhead led the authors to retain reward-weighted training as the main approach.

Experimental Setup

- ❖ **Offline evaluation:** Assess ranking quality using historical interaction data, checking whether GenRec ranks the items members actually engaged with near the top.
- ❖ **Online evaluation:** Measure performance with real Netflix members receiving recommendations, observing changes in engagement and the long-term core metric.
- ❖ **A/B test:** Compare groups receiving recommendations from either the production ranker (control) or GenRec (treatment) to estimate GenRec's effect on online metrics.
- ❖ **Mean Reciprocal Rank (MRR):** Measures how high a relevant item appears in a ranking. Its contribution is $1/r$, where r is its rank: positions 1, 2, and 5 contribute 1, 0.5, and 0.2. Averaging these values gives MRR; higher is better.

Results

- ❖ GenRec is compared against Netflix's mature production discriminative ranker, which has been tuned over many years.
- ❖ Offline: GenRec achieves approximately +1.6% relative improvement in Mean Reciprocal Rank (MRR) while using about 40× fewer Phase-2 labeled training examples and fewer input signals than the production model.
- ❖ Online: A large-scale A/B test covers approximately 10% of Netflix traffic for four weeks, focusing on key batch-compute recommendation surfaces.

Results

- ❖ Figure 3 reports significant improvements over production on both online metrics:
- ❖ Short-term homepage engagement: +0.115%, with $p = 3.1 \times 10^{-10}$
- ❖ Long-term core metric: +0.006% relative improvement, with $p = 0.025$.
- ❖ These results demonstrate the data efficiency of Phase-2 post-training; the 40× comparison does not include the foundation model’s earlier training.

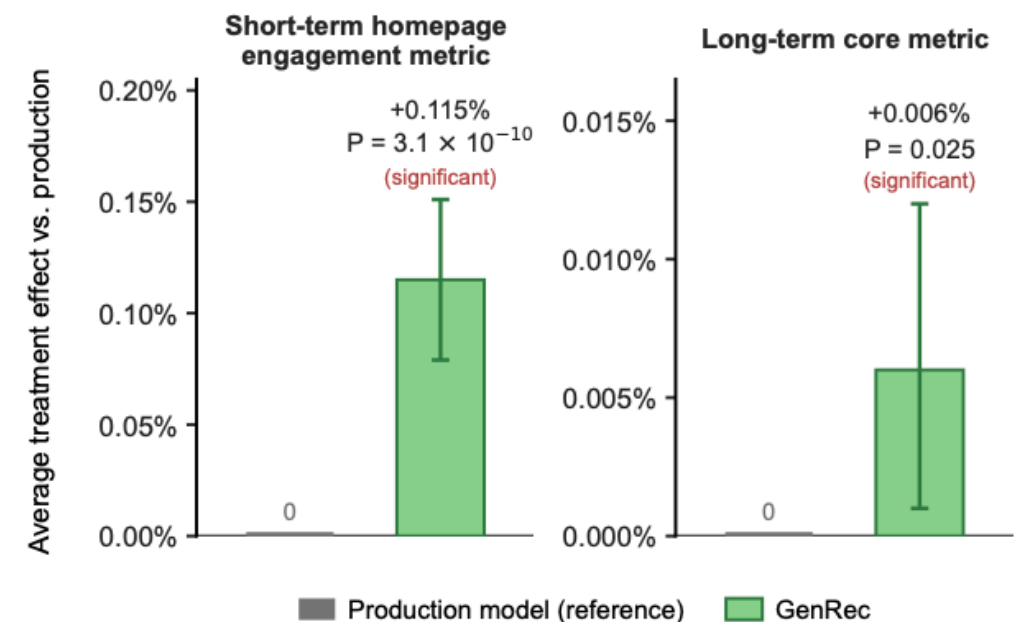


Figure 3: Online metrics of GenRec vs. production model. GenRec achieves statistically significant improvements on both short-term and long-term online metrics. ²

Results

- ❖ Increasing Phase-2 data from 1x to 20x consistently improves MRR for approximately 1-billion- and 10-billion-parameter models.
- ❖ Figure 4: The 10-billion-parameter model's normalized MRR increases from 1.00 to approximately 1.16.
- ❖ Larger models achieve higher MRR under the tested fixed training budget.

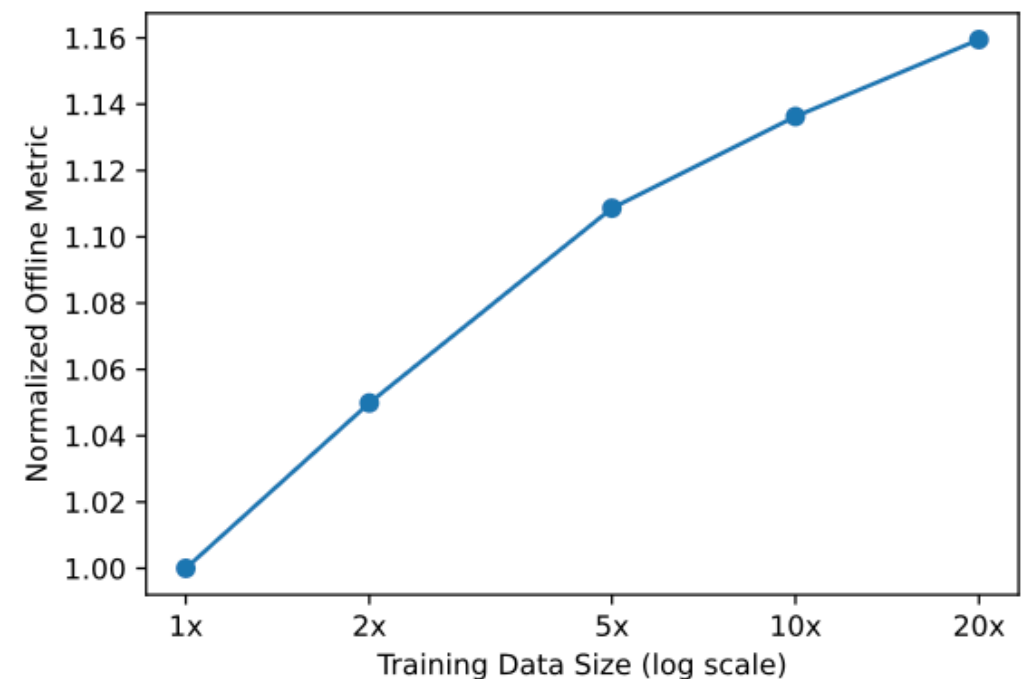


Figure 4: Phase-2 data scaling for the ~10B model.

Results

Table 1: Impact of Phase 1 vs. Phase 2 training

Offline ranking metrics (MRR)	
Impact of Phase 1	+ 10-20% compared to using an OSS model as base
Impact of Phase 2	+ 35-50% compared to a freshly trained Phase 1, with the gain increasing over time

- ❖ Phase 1: +10–20% in offline ranking metrics compared with an off-the-shelf LLM backbone.
- ❖ Phase 2: +35–50% over the fresh Phase-1 model.
- ❖ Phase-2 improvement reaches approximately 80% after two weeks, partly reflecting Phase-1 staleness.

Results

- ❖ History length: N events gives -7.9% ; $2N$ is the selected baseline; $3N$ gives only $+1.7\%$.
- ❖ Context is compressed from approximately 5,000 to 1,700 tokens, with negligible offline quality loss.
- ❖ Serving cost falls to approximately one-third of the original cost.

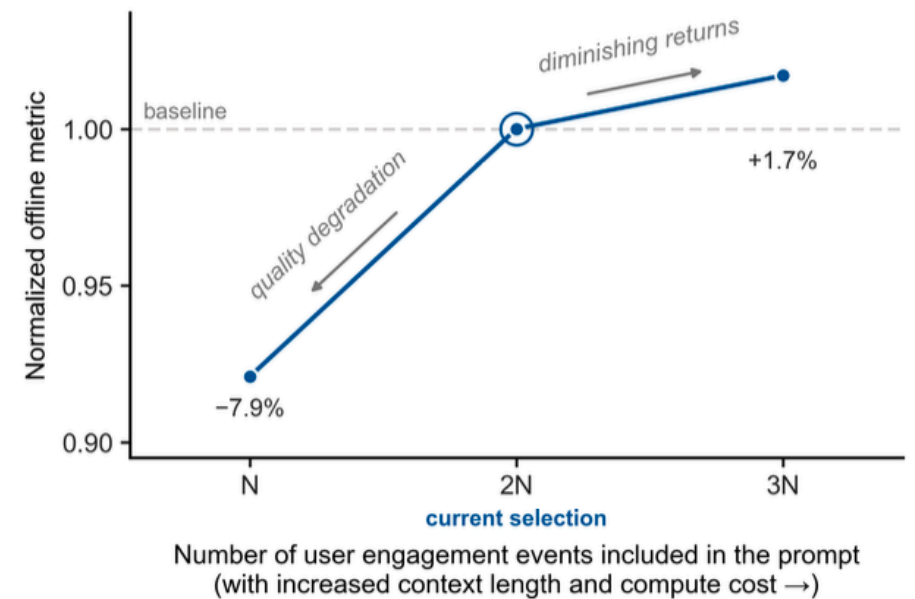


Figure 5: Normalized offline ranking metric (MRR) vs. number of user engagement events included in the prompt. The dashed line marks the baseline elbow point: increasing the number of events beyond this yields diminishing returns.

Thank you for your attention