Preprocessing Choices in Fine-Tuning Data Matter: Utility and Demographic Sensitivity in LLM-Based Recommendation

Furkan Yasir Goksu*
Department of Artificial Intelligence and Data Engineering
Istanbul Technical University
Istanbul, Türkiye
goksu20@itu.edu.tr

Yaren Yilmaz*
Department of Computer Engineering
Istanbul Technical University
Istanbul, Türkiye
yilmazy20@itu.edu.tr

Sule Gunduz Oguducu*
Department of Artificial Intelligence and Data Engineering
Istanbul Technical University
Istanbul, Türkiye
sgunduz@itu.edu.tr

Abstract

Large language models (LLMs) have introduced a generative approach to recommendation, producing item lists from textual representations of user histories and item information. However, this generative setting raises challenges concerning output validity, prompt consistency, and sensitivity to demographic descriptors. In this work, we investigate how preprocessing choices in fine-tuning data affect recommendation utility and fairness-related behavior in a generative LLM-based recommender system. We first establish a controlled recommendation pipeline that combines supervised fine-tuning, catalog-constrained decoding, and prompt augmentation. We then study three preprocessing choices: item-title representation, demographic balancing, and auxiliary item-knowledge enrichment. Experiments on MovieLens-1M show that the catalog-constrained pipeline eliminates out-of-catalog items and ensures complete ten-item outputs, while prompt augmentation improves prompt consistency and utility. Original MovieLens titles increase HR@10 from 10.67% to 12.47% and Coverage@10 from 41.36% to 47.88%. Fairness-related effects are attribute-dependent: age balancing and knowledge enrichment reduce selected group-level utility gaps while largely preserving utility, whereas gender balancing reduces utility without improving the evaluated gender-related indicators. Overall, preprocessing choices in fine-tuning data should be evaluated jointly in terms of utility, output behavior, and demographic sensitivity.

Keywords

LLM-based recommender systems, fine-tuning data preprocessing, fairness-related behavior, constrained decoding, demographic prompt sensitivity, catalog coverage

1 Introduction

Large language models (LLMs) are increasingly being used in recommender systems, enabling recommendations through natural language prompts [1], conversational interaction [2], textual item representations [3], and domain-specific fine-tuning [4]. Their understanding of natural language allows user histories, preferences, and candidate items to be represented in a single format. However, text-based recommendation introduces risks that are less apparent in conventional recommender systems [5], including malformed outputs, repeated or out-of-catalog items [6], and prompt sensitivity [7]. These challenges highlight the importance of understanding not only the quality of LLM-generated recommendations, but also the factors that shape their outputs.

Unlike conventional recommender systems that typically learn user and item representations from interaction data, LLM-based recommender systems can formulate recommendation tasks through natural language and textual representations of user interactions, item information, and recommendation instructions. In this setting, the way user preferences, interaction histories, and item information are expressed in the model input can directly influence the generated recommendations. This makes both the design of training data and the formulation of inference-time prompts important design choices. In generative LLM-based recommendation settings, the model generates a sequence of recommended items conditioned on a prompt that represents the user's interaction history and may also include demographic attributes, which are commonly treated as sensitive attributes in fairness research. This raises an important concern about whether LLM-based recommender systems can be sensitive to demographic information embedded in the model input. Although end users may not explicitly state demographic attributes in a recommendation request, such attributes may still be available to a recommender through user profiles or other contextual information. We therefore introduce demographic descriptors explicitly as a controlled intervention, allowing us to examine whether recommendation outputs change when demographic information is varied while the interaction history is held fixed.

Recent studies show that sensitive attributes can influence LLM recommendation outcomes, suggesting that fairness assessment in this setting requires more than standard accuracy metrics [8, 9]. Evaluation should therefore consider not only whether a model recovers held-out preferences, but also whether its outputs remain stable across demographic conditions and whether utility differs among user groups. At the same time, demographic sensitivity by itself does not necessarily constitute evidence of discrimination. We therefore interpret differences in recommendation outcomes across demographic conditions as fairness-related signals rather than direct evidence of discrimination [10, 11]. Such comparisons, however, require recommendation outputs that are sufficiently structured and consistent across conditions. LLM recommenders do not always provide this level of output control [5]. In particular, variations in output length or format can make it difficult to determine whether changes in recommendation lists are caused by preprocessing choices or by the generation process itself. This makes output

*Also with Artificial Intelligence and Data Science Application and Research Center, Istanbul Technical University.

3rd Workshop on Evaluating and Applying Recommender Systems with Large Language Models (EARL), Rome, Italy
2026.

control an important prerequisite for evaluating how preprocessing choices affect recommendation behavior.

This raises an important question of whether demographic sensitivity can be influenced before recommendation generation and, if so, which aspects of the recommendation pipeline affect it. Recent studies on LLM-based recommender systems have explored several adaptation strategies, including prompt-based recommendation, instruction tuning, hybrid textual and behavioral representations, data augmentation, and the use of external item knowledge [12, 13, 14, 15]. Together, these studies suggest that recommendation behavior can be shaped by choices made at different stages of the LLM recommendation pipeline. In parallel, fairness-oriented work has examined whether LLM recommenders produce different outcomes across sensitive user attributes and how such behavior can be mitigated through prompt- or inference-level interventions [8, 16, 17]. Despite these advances, comparatively less attention has been paid to the construction of fine-tuning data itself as a design factor and to how such choices may jointly affect recommendation utility and fairness-related behavior. In this work, we investigate three preprocessing choices in fine-tuning data, focusing on how item titles are represented, how demographic groups are distributed across the fine-tuning data, and whether additional item information is incorporated through auxiliary fine-tuning examples. We study these factors in the context of catalog-constrained LLM recommendation, without assuming that preprocessing alone resolves fairness concerns.

This study examines preprocessing in fine-tuning data as a design factor that can affect recommendation utility, output control, and demographic sensitivity together. We address four research questions. We first establish whether the generative recommendation pipeline provides a sufficiently controlled basis for comparison (RQ1), and then examine three preprocessing choices in fine-tuning data concerning item representation, demographic balancing, and auxiliary item knowledge (RQ2–RQ4).

- RQ1: Does the recommendation pipeline provide sufficient catalog validity, output control, and prompt consistency for controlled comparisons of preprocessing choices in fine-tuning data?
- RQ2: How does item-title representation affect recommendation utility and catalog usage within the catalog-constrained recommendation pipeline?
- RQ3: How does demographic balancing in fine-tuning data affect recommendation utility and sensitivity to demographic descriptors?
- RQ4: Can knowledge-enriched fine-tuning data reduce demographic sensitivity while preserving recommendation utility?

We investigate these questions by fine-tuning multiple LLM-based recommender variants within a shared catalog-constrained decoding framework. The variants differ only in three preprocessing choices in fine-tuning data: item-title representation, demographic balancing of recommendation examples, and auxiliary item-knowledge enrichment. We evaluate all variants under a shared protocol that combines utility with behavioral and fairness-oriented diagnostics. We measure whether generated lists recover held-out user preferences, remain valid and diverse within the target catalog, and change when demographic descriptors are added to the prompt. Evaluating these dimensions together reveals trade-offs that accuracy alone would miss, particularly when models achieve similar recommendation quality but exhibit different levels of demographic sensitivity.

Our findings demonstrate that preprocessing choices in fine-tuning data have heterogeneous effects on both utility and demographic sensitivity. Item-title representation substantially affects recommendation utility and catalog usage, while demographic balancing produces attribute-dependent outcomes: age balancing improves some fairness-related indicators while largely preserving recommendation utility, whereas gender balancing does not provide the same benefit. Auxiliary item-knowledge enrichment reduces the gender-related SNSR and narrows the neutral group-level utility gap for both attributes, although its effect on the remaining indicators is attribute-dependent. These interventions do not eliminate fairness concerns, but they provide empirical evidence that preprocessing choices in fine-tuning data can influence the trade-off between recommendation utility and fairness-related behavior.

The contributions of this work are as follows:

- We establish a controlled catalog-constrained LLM recommendation framework with prompt augmentation that provides a consistent experimental backbone for comparing preprocessing choices in fine-tuning data.
- We study three preprocessing choices in fine-tuning data: how items are presented, how demographic groups are distributed in the fine-tuning data, and whether additional item information is introduced through auxiliary fine-tuning examples. We analyze their effects on recommendation utility, group-level utility differences, and sensitivity to demographic descriptors, without introducing an explicit fairness-specific training objective.
- We employ a unified evaluation protocol for catalog-constrained LLM recommendation that jointly considers recommendation utility, output-control behavior, group-level utility differences, and demographic prompt sensitivity.
- We provide empirical evidence that preprocessing in fine-tuning data is not merely a technical preparation step, but a consequential design choice that can affect both recommendation performance and fairness-related behavior in LLM-based recommender systems.

2 Literature Review

Generative Recommendation and Catalog Grounding. Many LLM-based recommender systems represent users, items, collaborative interaction histories, and recommendation targets as textual sequences. Among the foundational approaches, P5 proposed a unified text-to-text framework for multiple recommendation tasks, followed by systems such as OpenP5 for developing and evaluating generative recommenders and GPTRec for autoregressive sequential recommendation, and GenRec, which formulates sequential recommendation as a sequence-to-sequence generation task [18, 19, 20, 21]. Subsequent work has further explored instruction tuning, collaborative signals, data augmentation, and external knowledge to adapt LLMs to recommendation [12, 13, 14, 22].

Nevertheless, compared with conventional fixed-candidate ranking, generative recommender systems face challenges such as malformed, repeated, or out-of-catalog items [5]. Catalog-grounding methods address these issues through retrieval, constrained generation, or item tokenization [6]. Within such frameworks, item representation is an important design choice, as alternative representation strategies can influence the way items are encoded and generated by the model [19]. Catalog constraints therefore provide a controlled output space for comparing alternative representation choices, although they do not themselves guarantee relevance, diversity, or fairness.

Prompt Sensitivity and Fairness. LLM-based recommendations can be sensitive to variations in prompts and generation, leading to inconsistent outputs for similar recommendation contexts. Recent studies show that variations in prompt formulation and repeated generation can affect recommendation outcomes [7, 23]. This variability is relevant to fairness evaluation, because recommendation changes associated with demographic information should be distinguished from general prompt sensitivity and output differences.

Fairness in recommendation concerns whether recommendation outcomes differ systematically across users or groups, particularly with respect to demographic attributes [10, 11]. In LLM-based recommendation, recent studies have examined whether sensitive user attributes influence generated recommendations and have proposed fairness-aware prompting and mitigation strategies [8, 9, 16, 17, 24]. Thus, differences in recommendation outcomes across demographic conditions should be interpreted under controlled prompt conditions and together with recommendation utility, rather than as direct evidence of unfairness.

Preprocessing Choices in Fine-Tuning Data. Data-level interventions have been explored in conventional recommender systems as a means of improving fairness. Chen et al. employ data augmentation to rebalance the training distribution, while Melchiorre et al. investigate resampling as a preprocessing strategy for mitigating gender-related disparities in recommendation outcomes [25, 26]. These studies demonstrate that the composition of training data can itself serve as an intervention point for fairness.

In LLM-based recommendation, prior work has examined aspects of fine-tuning data and task construction primarily from the perspectives of recommendation performance and efficiency. Lin et al. investigate representative sample selection for fine-tuning [27], while Liu et al. propose a user-level supervised fine-tuning (SFT) formulation for injecting collaborative information into an LLM-based sequential recommendation framework [28]. However, comparatively less attention has been paid to how preprocessing choices in LLM fine-tuning data affect both recommendation utility and sensitivity to demographic information.

To address this gap, this study examines three preprocessing choices in the fine-tuning data, namely item title representation, demographic balancing, and auxiliary item-knowledge enrichment, within a shared catalog-constrained framework.

3 Methodology

Figure 1 presents an overview of the controlled experimental pipeline used in this study. We first establish a shared catalog-constrained recommendation backbone and then study three preprocessing choices in fine-tuning data: item-title representation, demographic balancing, and auxiliary item-knowledge enrichment. We then evaluate recommendation utility, output behavior, prompt consistency, and demographic sensitivity.

3.1 Task Formulation

Let $\mathcal{U}$ denote a set of users and $\mathcal{I}$ a fixed item catalog. For each user $u \in \mathcal{U}$, we observe a chronologically ordered interaction history $H_u = (i_{u,1}, i_{u,2}, \ldots, i_{u,n_u})$, where n_u is the history length of user u and each $i_{u,j} \in \mathcal{I}$ is an item associated with a positive preference signal. The task is to generate an ordered recommendation list

$$R_u = (r_{u,1}, r_{u,2}, \ldots, r_{u,K})$$

conditioned on H_u, where each $r_{u,j} \in \mathcal{I}$ and the recommended items are expected to be distinct.

We formulate recommendation as conditional language generation. A prompt function ϕ converts a user history and an optional descriptor d into a textual input x_u, with $d = \varnothing$ denoting a neutral prompt. The generation process f_θ, comprising the fine-tuned model and its decoding procedure, produces a token sequence $\widehat{y}_u$, which is parsed into a catalog-item recommendation list $\widehat{R}_u$ by a representation-specific parser π_ρ:

$$x_u = \phi(H_u; d), \qquad \widehat{y}_u = f_\theta(x_u; \xi_u), \qquad \widehat{R}_u = \pi_\rho(\widehat{y}_u).$$

Here, ξ_u denotes decoding randomness when stochastic decoding is used, and ρ denotes the item-representation configuration; for readability, the dependence of the decoding procedure on ρ is suppressed in the notation. Within $\widehat{y}_u$, each item span is delimited by <SOI> and <EOI> and represents a single serialized catalog item. Recommendation metrics are computed on $\widehat{R}_u$, while structural deviations such as repeated items are retained as observable output behavior and evaluated separately.

Each supervised recommendation example consists of an input prompt x_u and a serialized target sequence $y_u = (y_{u,1}, \ldots, y_{u,L_u})$, where L_u denotes its token length.

An item-representation configuration ρ defines a serialization function s_ρ that determines how catalog items are represented in the textual recommendation pipeline, including training targets, parser mappings, and decoding constraints. Let $\tau_\rho(i)$ denote the token sequence obtained by tokenizing $s_\rho(i)$ under the tokenizer and formatting convention used within an item span. Changing s_ρ leaves the underlying catalog unchanged but may alter tokenization, prefix sharing, and the set of token sequences the language model can generate for catalog items. To ensure that π_ρ maps generated item spans back to catalog items unambiguously, serialized representations must uniquely identify items in the catalog under evaluation; representation-induced collisions are therefore resolved before training and evaluation.

We study three preprocessing choices in fine-tuning data: item representation, demographic balancing, and auxiliary item knowledge. Demographic balancing modifies the sampling distribution of fine-tuning examples without introducing a fairness-specific objective, whereas auxiliary item knowledge adds supplementary item-information examples to the fine-tuning corpus.

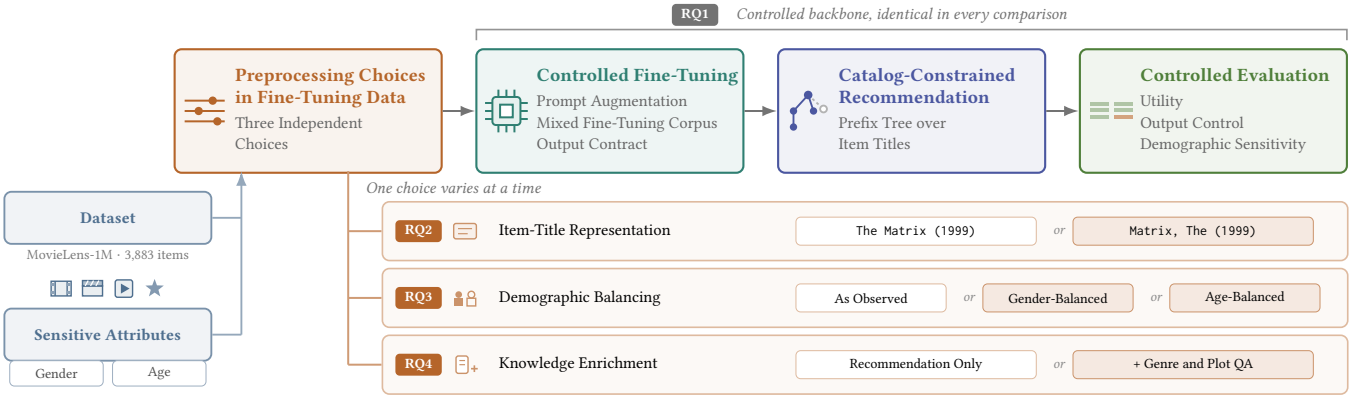


Figure 1: Overview of the controlled experimental pipeline.

3.2 Training Objective and Output Contract

The model is trained to generate recommendation lists using explicit item-span markers. Each catalog item in the target sequence is serialized as

$$\texttt{<SOI>}\ s_\rho(i)\ \texttt{<EOI>}.$$

The start-of-item (SOI) marker indicates the beginning of a catalog item span, while the end-of-item (EOI) marker closes the span. This output contract provides a deterministic parsing rule for well-formed outputs and defines the interface between language generation and catalog-constrained decoding.

The model is trained using scope-masked cross-entropy. Outside item spans, the objective is equivalent to ordinary autoregressive cross-entropy, whereas within item spans the candidate vocabulary is restricted to catalog-valid continuations when multiple valid continuations are available. Let

$$C_\rho = \{\tau_\rho(i) : i \in \mathcal{I}\}$$

denote the tokenized catalog, and let $\mathcal{V}$ denote the model token vocabulary. For a target position ℓ inside an item span, let $q_{u,\ell}$ denote the target item-token prefix following the most recent <SOI> and preceding position ℓ. The set of valid next tokens for a prefix q is

$$A_\rho(q) = \{v \in \mathcal{V} : \exists s \in C_\rho,\ q \,\|\, (v) \preceq s\} \ \cup\ \{\texttt{<EOI>} : q \in C_\rho\},$$

where $\|$ denotes sequence concatenation and $\preceq$ denotes the sequence-prefix relation. The closing marker is therefore available only when the current prefix corresponds to a complete catalog item, including cases in which that item is also a prefix of a longer catalog representation.

For the purpose of defining the candidate vocabulary, the closing marker is treated as part of the item span. The candidate vocabulary at target position ℓ is then defined as

$$S_{u,\ell} = \begin{cases} A_\rho(q_{u,\ell}), & \text{if } \ell \text{ is within an item span and } |A_\rho(q_{u,\ell})| > 1, \\ \mathcal{V}, & \text{otherwise.} \end{cases}$$

The resulting training objective is

$$\mathcal{L}_{\text{scope}}(u) = -\sum_{\ell=1}^{L_u} \log \frac{p_\theta(y_{u,\ell} \mid x_u, y_{u,<\ell})}{\sum_{v \in S_{u,\ell}} p_\theta(v \mid x_u, y_{u,<\ell})}.$$

When $S_{u,\ell} = \mathcal{V}$, the denominator equals one and the objective reduces to standard autoregressive cross-entropy. At positions within item spans where multiple catalog-valid continuations are available, the loss is normalized over those valid continuations. If an item prefix has only one valid continuation, the full vocabulary is retained to avoid a one-candidate softmax that would otherwise contribute zero training loss.

3.3 Catalog-Constrained Generation

Unconstrained LLM-based recommenders may generate plausible items that are not part of the target catalog. Following recent work on eliminating out-of-catalog recommendations in LLM-based recommender systems [6], we use prefix-tree constrained decoding to ground item generation in the fixed catalog.

For a given item representation ρ, the token sequences in C_ρ are inserted into a prefix tree, where each completed path corresponds to a serialized catalog item. When the model emits <SOI>, decoding enters catalog-constrained mode. At generation position ℓ, let $\widehat{q}_{u,\ell}$ denote the generated within-item prefix following the most recent <SOI> and preceding position ℓ. The next-token distribution is restricted to the valid continuation set $A_\rho(\widehat{q}_{u,\ell})$:

$$\widetilde{p}_\theta\left(v \mid x_u, \widehat{y}_{u,<\ell}\right) = \frac{p_\theta\left(v \mid x_u, \widehat{y}_{u,<\ell}\right) \mathbf{1}[v \in A_\rho(\widehat{q}_{u,\ell})]}{\sum_{w \in A_\rho(\widehat{q}_{u,\ell})} p_\theta\left(w \mid x_u, \widehat{y}_{u,<\ell}\right)}.$$

The <EOI> marker becomes available only when the current prefix corresponds to a complete catalog-item sequence. Emitting this marker closes the item span and returns decoding to the ordinary generation mode. Outside item spans, generation proceeds over the full model vocabulary.

Catalog-constrained decoding removes out-of-catalog generation as a confounding factor by preventing invalid catalog strings within item spans, but it does not guarantee recommendation relevance, diversity, or fairness.

Because the prefix tree is constructed from the representation-specific catalog sequences C_ρ, item representation determines both the training target space and the constrained generation space. Figure 2 illustrates how valid and invalid continuations are handled within an item span.

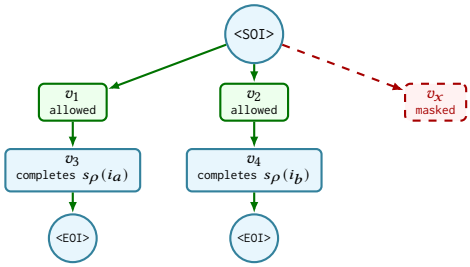


Figure 2: Prefix-tree constrained decoding within an item span. After the model emits <SOI>, only tokens that continue a valid catalog path remain available. Invalid continuations are masked, and <EOI> becomes available only after a complete catalog-item sequence has been generated.

The decoder does not explicitly enforce exactly K completed item spans, prevent early sequence termination, or mask previously generated items. Therefore, Exact-K compliance and repetition are treated as empirical output-control measures rather than guaranteed properties of constrained decoding.

3.4 Prompt Augmentation

Language model behavior can change across instruction paraphrases, prompt templates, and small textual perturbations, even when the task itself is unchanged [29, 30]. A model fine-tuned on a single instruction wording may therefore become dependent on that particular presentation. To reduce this dependence, for prompt-augmented configurations, we expose the model during fine-tuning to multiple prompt formulations of the recommendation instruction. The goal is to encourage robustness to changes in prompt wording and presentation format while preserving the underlying recommendation task.

Prompt Perturbation Consistency Learning (PPCL) addresses prompt sensitivity by introducing a consistency objective between clean and perturbed prompts [31]. Inspired by this idea, we use template-based prompt augmentation, but incorporate the augmented prompts directly into fine-tuning without an additional consistency objective.

Formally, we extend the prompt function introduced in the task formulation from a single function ϕ to a collection of prompt templates $\Phi = \{\phi_1, \phi_2, \ldots, \phi_M\}$. For a fixed user history H_u and descriptor d, template $\phi_m \in \Phi$ constructs the input

$$x_u^{(m)} = \phi_m(H_u; d).$$

During prompt-augmented fine-tuning, we set $d = \varnothing$. The templates vary the instruction wording, constraint emphasis, preference framing, prompt format and length, and minor text perturbations, while preserving the underlying interaction history, serialized target sequence, item representation, and recommendation objective.

In addition to prompt instruction variation, we also vary how the interaction history H_u is presented in the prompt without changing its item content or ordering. During construction of a supervised recommendation example, a prompt template and one of the supported history formats are selected to form the input. For a fixed example and item-representation configuration ρ, the target sequence y_u, catalog $\mathcal{I}$, serialization function s_ρ, and output contract remain unchanged. Thus, prompt augmentation changes only how the input is expressed, while preserving the recommendation task and its target space.

In contrast to training-time prompt augmentation, attribute-conditioned prompting is used during evaluation to measure sensitivity to demographic descriptors while keeping the underlying recommendation context fixed.

3.5 Attribute-Conditioned Prompt Evaluation

To evaluate fairness-related behavior, we compare recommendations generated from the same interaction history under neutral and attribute-conditioned prompt settings. Let a denote a user attribute and let $g \in \mathcal{G}_a$ denote one of its possible values or groups. For a fixed evaluation prompt function ϕ_{eval} and user history H_u, the neutral and attribute-conditioned prompts are defined as

$$x_u^0 = \phi_{\text{eval}}(H_u; \varnothing), \qquad x_u^{a,g} = \phi_{\text{eval}}(H_u; (a, g)).$$

The interaction history, recommendation instruction, prompt template, decoding configuration, and target catalog are held fixed across the paired conditions; only the injected attribute descriptor changes. Under stochastic decoding, the same decoding randomness ξ_u is used for both conditions to reduce variation unrelated to the attribute intervention. The corresponding recommendation lists are

$$\widehat{R}_u^0 = \pi_\rho\left(f_\theta(x_u^0; \xi_u)\right), \qquad \widehat{R}_u^{a,g} = \pi_\rho\left(f_\theta(x_u^{a,g}; \xi_u)\right).$$

Differences between $\widehat{R}_u^0$ and $\widehat{R}_u^{a,g}$ are treated as attribute-sensitivity signals, since only the attribute descriptor varies across the paired conditions. These comparisons are aggregated across users for each evaluated attribute and group.

Each pair yields a set overlap between $\widehat{R}_u^0$ and $\widehat{R}_u^{a,g}$, which we report as a Jaccard similarity and average within each group g. The spread of these group averages across $\mathcal{G}_a$ is reported as SNSR, and is small when the descriptor shifts every group by a similar amount. Group-level utility differences are measured separately under the neutral prompt and reported as the Neutral Gap. Section 4.3 gives the definitions.

This protocol measures fairness-related behavioral variation rather than fairness in an absolute sense. Such differences do not by themselves constitute evidence of discrimination and are therefore interpreted together with recommendation utility and other behavioral measures.

4 Experimental Setup

This section describes the dataset, chronological split, and preparation of fine-tuning data, followed by the implementation details and evaluation metrics used across the experiments.

4.1 Dataset, Chronological Split, and Fine-Tuning Data Preparation

We conduct our experiments on MovieLens-1M, which contains 6,040 users, 3,883 movies, and 1,000,209 ratings [32]. Ratings greater than or equal to 4 are treated as positive interactions. Users with fewer than 22 positive interactions are excluded, ensuring sufficient

interactions for the maximum-length training instance of ten history items and ten targets while reserving one interaction each for validation and testing. These criteria retain 4,995 users. For each user, positive interactions are ordered chronologically; the final interaction is separated as the test target, the preceding interaction is used for validation, and the remaining earlier interactions are used for the training set. This split yields one validation target and one test target for each of the 4,995 retained users. The recommendation fine-tuning records are then sampled from the remaining training interactions, as described below. The full catalog of 3,883 movies is retained for constrained generation; 3,432 items (88.4%) occur in the training interactions, while 451 items (11.6%) remain unseen in the recommendation fine-tuning data.

Recommendation fine-tuning examples are constructed as history-to-target generation instances. For each user, we sample 20 training instances from the chronological training sequence, yielding 99,900 recommendation examples in total. Each instance contains a variable-length history and an ordered target list, with both lengths capped at ten items. The requested target count K is written into the instruction, so the prompt and the target sequence always agree on list length. Extremely long raw titles are filtered using a 99th-percentile title-length cutoff to avoid formatting outliers. Histories are serialized using comma-separated, arrow-separated, semicolon-separated, and numbered-list formats while preserving chronological item order.

We evaluate two item-title representations. The normalized-title condition moves trailing articles to the beginning of the title, e.g., *Matrix, The (1999)* becomes *The Matrix (1999)*. The original-title condition preserves the MovieLens surface form, e.g., *Matrix, The (1999)*. This distinction affects the token-level structure of the catalog: normalization rewrites 714 of the 3,883 titles, and the resulting catalog places 715 titles, or 18.4% of the catalog, behind one shared opening token. Under the original MovieLens form, no opening token accounts for more than 1.8% of the catalog. Both representations are applied consistently to training targets, parsing, and prefix-tree constrained decoding.

Following the RecLM-cgen design, which augments recommendation fine-tuning with conversational data to help preserve general conversational ability [6], we construct a mixed fine-tuning corpus containing 119,056 examples: 99,900 recommendation examples (83.9%), 14,161 general conversational examples (11.9%), and 4,995 recommendation-oriented conversational examples (4.2%). These proportions are design choices guided by prior mixed fine-tuning practice rather than fixed values copied from prior work.

To reduce dependence on a single prompt wording, recommendation examples are expressed using 37 semantically equivalent instruction templates organized into six families as given in Table 1. The templates differ in phrasing rather than task semantics, spanning direct recommendation, explicit rule lists, preference framing, output-format emphasis, compressed phrasing, and minor misspellings. The same prompt-augmentation strategy is applied across the compared fine-tuning configurations. For example, a T6 template replaces *Recommend* with *Recomend* while keeping the history, requested list length, and output requirements unchanged.

For the demographic-balancing conditions used in RQ3, we modify only the recommendation fine-tuning examples. The recommendation corpus size is held fixed at 99,900 examples, and the two groups of the balanced attribute receive 49,950 examples each. Under gender balancing, 3,620 male users contribute 13.8 examples each on average and 1,375 female users contribute 36.3. Under age balancing, using the binary split at 35, 2,152 older users contribute 23.2 examples each and 2,843 younger users contribute 17.6. Demographic labels are used only for constructing these balanced corpora and are not included in the recommendation prompts.

Table 1: Prompt template families used during fine-tuning. Shares give the proportion of recommendation examples assigned to each family.

Family	Character	Templates	Share
T1	Direct request	8	30%
T2	Explicit rule list	7	25%
T3	Preference framing	7	20%
T4	Output format emphasis	6	15%
T5	Compressed phrasing	5	7%
T6	Misspelled instructions	4	3%
Total		37	100%

For the knowledge-enriched condition used in RQ4, we add 7,683 auxiliary item-level question–answer examples about catalog movies. These examples cover genre and plot-style knowledge: genres are taken from MovieLens metadata, and plot summaries are taken from TMDB and joined to the catalog by item identifier. Answers that name a movie use the same <SOI> and <EOI> item-span markers as the recommendation task. The knowledge examples are added to the fine-tuning corpus rather than replacing recommendation examples, resulting in 126,739 total fine-tuning examples.

4.2 Implementation Details

All configurations fine-tune Llama-3-8B-Instruct [33]. The base weights are loaded in 4-bit NF4 with double quantization and a bfloat16 compute type, and low-rank adapters are trained on top, following QLoRA [34, 35]. LoRA adapters are applied to the attention query, key, value, and output projections and to the feed-forward gate, up, and down projections. We use rank $r = 16$, scaling factor $\alpha = 32$, and dropout 0.05. Because <SOI> and <EOI> are added as new tokenizer entries, the input embedding matrix and output language-model head are trained alongside the adapters.

Optimization uses paged 8-bit AdamW with a learning rate of 10^{-4}, a cosine schedule, and a warmup ratio of 0.03. Weight decay is set to zero and gradients are clipped at a norm of 1.0. Attention is computed with FlashAttention-2, and training uses a device batch of one sequence and gradient accumulation over 16 steps, giving an effective batch size of 16. Training performs a single pass over the fine-tuning corpus, so each example is seen exactly once.

At inference time, models are served with vLLM [36]. All headline evaluations use greedy decoding at temperature zero, a context window of 1,024 tokens, and a generation budget of 512 new tokens. Item spans are decoded with the prefix-tree constraint unless an unconstrained ablation is explicitly reported. Each test user is evaluated with the same ten-item history and a request for ten recommendations. Each configuration is based on a single fine-tuning

run with fixed data-generation and training seeds and is evaluated deterministically using a fixed checkpoint and prompt set.

4.3 Evaluation Metrics

We report metrics covering five complementary dimensions: recommendation utility, catalog usage, output control, prompt consistency, and demographic sensitivity. Recommendation utility is measured with Hit Ratio@10 (HR@10), which indicates whether the held-out test item appears among the ten generated recommendations. Catalog usage is summarized with Coverage@10 and Unique Items. Coverage@10 is the proportion of catalog items that appear at least once across all generated top-10 lists, while Unique Items reports the corresponding count of distinct recommended catalog items.

Output control is evaluated with Exact-10, Out of Domain@10 (OOD@10), and Repeat@10. Exact-10 is the proportion of responses containing exactly ten extracted item spans. OOD@10 measures the proportion of generated recommendations that do not correspond to valid catalog items. Repeat@10 measures within-list duplication among the generated recommendations. Higher Exact-10 and lower OOD@10 and Repeat@10 indicate more reliable recommendation outputs.

Prompt consistency is evaluated in RQ1 using a separate set of five evaluation prompt variants. The variants preserve the same user history, target list length, and output contract while changing the request wording through a clean reference, paraphrase, instruction-wording change, format edit, and compact prompt. Prompt HR Spread is the difference between the highest and lowest HR@10 across the five variants. Prompt Jaccard compares each non-reference variant with the clean reference prompt for the same users and averages the resulting top-10 set overlaps across users and variants.

Fairness-related behavior is evaluated through paired prompt comparisons. For each user, we compare recommendations generated from the same interaction history under a neutral prompt and a demographic-conditioned prompt. Neutral-conditioned Jaccard measures the overlap between these two top-10 recommendation sets; higher values indicate that adding a demographic descriptor changes the recommendation list less. Pair Jaccard, used in RQ3, directly compares recommendation sets generated for the same history under two demographic descriptor values, such as male versus female or age < 35 versus age ≥ 35.

We also report Neutral Gap and Sensitive-to-Neutral Similarity Range (SNSR) [8]. Neutral Gap is the absolute difference in HR@10 between demographic groups under the neutral prompt, reported in percentage points. For a demographic attribute with groups $\mathcal{G}_a$, let $\overline{J}_g$ denote the average neutral-conditioned Jaccard similarity for users in group g. SNSR is defined as

$$\mathrm{SNSR} = \max_{g \in \mathcal{G}_a} \overline{J}_g - \min_{g \in \mathcal{G}_a} \overline{J}_g.$$

Lower SNSR indicates that the demographic descriptor affects the evaluated groups more uniformly. We interpret Neutral Gap, Neutral-conditioned Jaccard, Pair Jaccard, and SNSR as indicators of demographic sensitivity and group-level utility disparity, not as direct evidence of discrimination.

5 Experimental Results

We organize the experimental analysis around four research questions that examine the reliability of the recommendation pipeline and the effects of preprocessing choices in fine-tuning data. We first establish the controlled recommendation backbone and then evaluate item-title representation, demographic balancing, and additional item information.

5.1 RQ1: Does the recommendation pipeline provide sufficient catalog validity, output control, and prompt consistency for controlled comparisons of preprocessing choices in fine-tuning data?

Generative recommenders can return lists of the wrong length, repeat items, or name items that are not in the catalog [5, 6]. RQ1 therefore examines whether the pipeline provides a sufficiently controlled basis for the later comparisons of preprocessing choices and demographic sensitivity. Recommendation utility is part of that basis. A model whose lists are only weakly tied to the input can appear insensitive to demographic descriptors simply because it recommends poorly, so stability under demographic prompting is informative only once the pipeline recovers held-out preferences at a reasonable rate. We first evaluate a structured supervised fine-tuning (SFT) baseline under unconstrained decoding and then apply prefix-tree constrained decoding to the same SFT weights to isolate the effect of catalog grounding. We next introduce the catalog-constrained training configuration and finally add prompt augmentation while keeping the catalog-constrained generation mechanism fixed. Prompt consistency is evaluated separately using five evaluation prompt variants. Table 2 summarizes these configurations.

The SFT baseline learns the structured `<SOI>–<EOI>` output format, but SFT alone does not provide reliable list-level control. Although it achieves an HR@10 of 7.47%, only 0.06% of its outputs contain exactly ten recommendations, and 2.08% of its recommended items fall outside the catalog. This indicates that learning the serialization format alone does not prevent variable-length or out-of-catalog outputs.

Applying the prefix tree to the same SFT weights (SFT constrained decoding) reduces OOD@10 from 2.08% to 0.00% while leaving HR@10 essentially unchanged (7.47% versus 7.43%). This comparison isolates the effect of catalog grounding and shows that constrained decoding can eliminate invalid catalog items without substantially changing recommendation utility. However, constrained decoding alone does not provide list-level control: Exact-10 remains near zero (0.02%), while Repeat@10 increases from 3.12% to 4.86%.

The catalog-constrained training configuration improves list-level consistency. Exact-10 increases to 100%, OOD@10 remains at 0.00%, and HR@10 improves to 9.89%. This configuration produces consistently sized recommendation lists while restricting generated item spans to catalog-valid paths, thereby providing a more controlled output space for subsequent comparisons. However, repeated-item behavior remains substantial, with Repeat@10 reaching 10.56%. Thus, catalog validity and exact-length compliance do not by themselves eliminate within-list redundancy.

The prompt-augmented model retains the same catalog-constrained decoding mechanism while exposing the model to multiple semantically equivalent recommendation instructions during fine-tuning. The resulting model preserves 100% Exact-10 compliance and zero out-of-catalog generation, while improving HR@10 from 9.89% to 10.67% and reducing Repeat@10 from 10.56% to 8.01%. Prompt augmentation also reduces sensitivity to prompt formulation: Prompt HR Spread decreases from 2.18 to 0.26 percentage points, while Prompt Jaccard increases from 0.6799 to 0.7489. These results indicate that prompt-augmented fine-tuning improves recommendation utility and prompt-level consistency while preserving the structural control established by the catalog-constrained pipeline.

Taken together, these results show that SFT learns the structured recommendation format but does not provide reliable list-level control. Prefix-tree constrained decoding eliminates out-of-catalog generation, while the catalog-constrained training configuration provides consistent list-level outputs. Prompt augmentation preserves this structural reliability while improving recommendation utility, reducing repeated-item behavior, and decreasing sensitivity to prompt formulation. We therefore use the catalog-constrained, prompt-augmented pipeline as the controlled backbone for the remaining preprocessing and demographic-sensitivity experiments.

5.2 RQ2: How does item-title representation affect recommendation utility and catalog usage within the catalog-constrained recommendation pipeline?

RQ2 examines whether the textual representation of catalog items affects recommendation utility and catalog usage while the underlying item identities remain unchanged. The prompt-augmented configuration uses normalized titles, where articles stored at the end of MovieLens titles are moved to the beginning; for example, *American President, The (1995)* becomes *The American President (1995)*. The original-title condition preserves the MovieLens title format. Both conditions use the same underlying catalog and recommendation setting.

Table 3 shows higher recommendation utility and broader catalog usage with the original MovieLens titles. HR@10 increases from 10.67% to 12.47%, while Coverage@10 rises from 41.36% to 47.88%. The number of unique recommended items also increases from 1,606 to 1,859. Repeat@10 changes only slightly, from 8.01% to 8.43%.

One possible explanation for these differences is the token-prefix structure induced by the two title representations. Moving common articles to the beginning can cause more normalized titles to share high-frequency initial tokens, increasing prefix sharing at early levels of the catalog trie; *The* is a common example. In contrast, the original MovieLens representation distributes titles across a wider set of initial prefixes. This changes the distribution of early prefixes and, consequently, the branching structure of the catalog prefix tree used during constrained decoding. Prior work also reported that textual item representation, and the resulting trie structure, can affect constrained generation and recommendation performance [6, 37]. However, changing the title representation also changes the token sequences used during fine-tuning. Therefore, the observed differences cannot be attributed to the prefix-tree structure alone.

Overall, RQ2 shows that alternative title representations of the same catalog are not interchangeable. Even when the underlying item identities remain unchanged, token ordering affects recommendation utility, catalog coverage, and the range of items recommended. Therefore, title formatting should be treated as a preprocessing choice to report explicitly rather than an implementation detail.

5.3 RQ3: How does demographic balancing in fine-tuning data affect recommendation utility and sensitivity to demographic descriptors?

RQ3 examines how balancing the fine-tuning data with respect to gender or age affects recommendation utility and fairness-related behavior. Both balancing interventions are evaluated under the same original-title representation. We report HR@10 to measure recommendation utility and Neutral Gap to measure the difference in HR@10 between demographic groups under the neutral prompt. Demographic sensitivity is evaluated from three complementary perspectives. Neutral-conditioned Jaccard measures the overlap between recommendations generated with the neutral prompt and the corresponding demographic-conditioned prompt. SNSR measures how much this neutral-to-conditioned similarity differs between the demographic groups. Pair Jaccard directly compares recommendations generated for the same interaction history under the two values of the evaluated demographic descriptor, such as male versus female or age < 35 versus age ≥ 35. Higher Jaccard values indicate greater recommendation stability under demographic prompting, while lower Neutral Gap and SNSR indicate smaller group-level utility differences and smaller differences in demographic sensitivity, respectively. Table 4 reports the results for both balancing interventions.

For gender, balancing decreases HR@10 from 12.47% to 11.32%. The Neutral Gap increases from 0.45 to 1.21 percentage points, indicating a larger difference in recommendation utility between the gender groups under the neutral prompt. Neutral-conditioned Jaccard decreases from 0.8312 to 0.8240, while SNSR increases from 0.0519 to 0.0636. Pair Jaccard also decreases, from 0.8110 to 0.7918. Across the reported metrics, gender-balanced fine-tuning therefore does not reduce the observed gender-related differences or sensitivity and is accompanied by lower overall recommendation utility.

The age-balanced configuration shows a different pattern. HR@10 changes only slightly, from 12.47% to 12.39%. The Neutral Gap decreases from 0.69 to 0.38 percentage points, while Neutral-conditioned Jaccard increases from 0.8195 to 0.8305. Pair Jaccard also increases, from 0.8407 to 0.8471. These changes indicate a smaller neutral-prompt utility difference between the age groups and greater recommendation overlap under both neutral-to-conditioned and direct descriptor-pair comparisons. However, SNSR increases from 0.0017 to 0.0077, showing that the neutral-to-conditioned similarity becomes somewhat less uniform across the two age groups. Age balancing therefore improves several of the

Table 2: Pipeline reliability, recommendation quality, and prompt consistency. Arrows give the preferred direction.

Model	HR@10 ↑	Exact-10 ↑	OOD@10 ↓	Repeat@10 ↓	Prompt HR Spread ↓	Prompt Jaccard ↑
SFT (unconstrained decoding)	7.47	0.06	2.08	3.12	–	–
SFT (constrained decoding)	7.43	0.02	0.00	4.86	–	–
Catalog-constrained	9.89	100.00	0.00	10.56	2.18	0.6799
Prompt-augmented	**10.67**	100.00	0.00	**8.01**	**0.26**	**0.7489**

Table 3: Effect of item-title representation on recommendation utility and catalog usage. Arrows give the preferred direction.

Model	HR@10 ↑	Coverage @10 ↑	Repeat @10 ↓	Unique Items ↑
Normalized titles	10.67	41.36	**8.01**	1,606
Original titles	**12.47**	**47.88**	8.43	**1,859**

Table 4: Effect of gender- and age-balanced fine-tuning on recommendation utility and demographic sensitivity under the original-title representation. Arrows give the preferred direction.

Model	HR@10 ↑	Neutral Gap ↓	Neutral-cond. Jaccard ↑	SNSR ↓	Pair Jaccard ↑
Gender					
Baseline	**12.47**	**0.45**	**0.8312**	**0.0519**	**0.8110**
+ Gender-balanced	11.32	1.21	0.8240	0.0636	0.7918
Age					
Baseline	**12.47**	0.69	0.8195	**0.0017**	0.8407
+ Age-balanced	12.39	**0.38**	**0.8305**	0.0077	**0.8471**

reported age-related indicators while not improving all sensitivity measures.

Taken together, the results show that the effect of demographic balancing depends on the demographic attribute and the evaluation measure considered. Gender balancing reduces overall recommendation utility and does not improve any of the reported gender-related indicators. In contrast, age balancing largely preserves overall recommendation utility while reducing the neutral age-group utility gap and increasing both Neutral-conditioned and Pair Jaccard. At the same time, its higher SNSR shows that the improvement is not consistent across all sensitivity measures. These results indicate that demographic balancing should be evaluated separately for each demographic attribute rather than assumed to have a uniform effect on fairness-related recommendation behavior.

5.4 RQ4: Can knowledge-enriched fine-tuning data reduce demographic sensitivity while preserving recommendation utility?

RQ4 examines how incorporating additional item information through auxiliary fine-tuning examples affects recommendation utility and demographic sensitivity. We evaluate knowledge enrichment using the same original-title backbone and the same core evaluation dimensions as in RQ3: HR@10, Neutral Gap, Neutral-conditioned Jaccard, and SNSR. The knowledge-enriched condition adds auxiliary examples to the fine-tuning corpus rather than replacing recommendation examples; consequently, this condition also contains more fine-tuning examples than the corresponding baseline.

Table 5: Effect of knowledge enrichment on recommendation utility and demographic sensitivity. Arrows give the preferred direction.

Model	HR@10 ↑	Neutral Gap ↓	Neutral-cond. Jaccard ↑	SNSR ↓
Gender				
Baseline	**12.47**	0.45	**0.8312**	0.0519
+ Knowledge	12.24	**0.10**	0.8270	**0.0291**
Age				
Baseline	**12.47**	0.69	0.8195	**0.0017**
+ Knowledge	12.24	**0.32**	**0.8201**	0.0123

Table 5 shows that knowledge enrichment does not substantially harm recommendation utility. Overall HR@10 decreases by only 0.23 percentage points, indicating that the main recommendation performance is largely preserved.

Knowledge enrichment reduces the Neutral Gap for both gender and age, while Neutral-conditioned Jaccard remains nearly unchanged. Thus, the smaller utility gaps are not accompanied by greater similarity between neutral and attribute-conditioned recommendation lists. This suggests that knowledge enrichment improves group-level utility alignment while leaving neutral-to-conditioned list overlap largely stable. However, the effect on SNSR differs across attributes. Gender SNSR decreases from 0.0519 to 0.0291, whereas age SNSR increases from 0.0017 to 0.0123. Although SNSR for the age attribute remains small, the opposite directions show that knowledge enrichment does not affect between-group differences in demographic sensitivity uniformly across attributes.

Overall, knowledge enrichment in fine-tuning data largely preserves recommendation utility while reducing group-level utility gaps for both gender and age. Its effect on SNSR is attribute-dependent: it lowers gender-related sensitivity but raises the age-related value, which nevertheless remains small in absolute terms. As in RQ3, the fairness-related effect of this preprocessing choice should therefore be interpreted separately for each attribute rather than assumed to be uniform. These findings suggest that incorporating additional item information into fine-tuning data can serve as a complementary data-level mechanism for reducing some forms of demographic sensitivity, even without an explicit fairness-specific training objective.

6 Conclusion

In this study, we examined how preprocessing choices in fine-tuning data affect utility and fairness-related behavior in generative LLM-based recommendation. Catalog-constrained decoding removed out-of-catalog items, the catalog-constrained training configuration produced complete ten-item outputs, and prompt augmentation improved prompt consistency and utility. Item-title representation affected both utility and catalog coverage even though the underlying item identities remained unchanged. The effects of demographic balancing and auxiliary item-knowledge enrichment were attribute-dependent. Age balancing reduced the age-related utility gap while largely preserving utility, whereas gender balancing reduced utility without improving the evaluated gender-related indicators. Knowledge enrichment preserved utility and reduced group-level gaps, but its effect on demographic sensitivity varied across attributes and measures. A preprocessing choice therefore cannot be judged on utility alone, nor on a single attribute. Future work should control the size of the knowledge-enriched corpus and extend the evaluation beyond one dataset and binary demographic groups.

Acknowledgments

This work is partially supported by the Scientific Research Project Unit of Istanbul Technical University under project number: 46051.

References

[1] Yueqing Liang, Liangwei Yang, Chen Wang, Xiongxiao Xu, Philip S Yu, and Kai Shu. 2025. Taxonomy-guided zero-shot recommendations with LLMs. In *Proceedings of the 31st International Conference on Computational Linguistics*, 1520–1530.

[2] Chuang Li, Yang Deng, Hengchang Hu, Min-Yen Kan, and Haizhou Li. 2025. ChatCRS: incorporating external knowledge and goal guidance for LLM-based conversational recommender systems. In *Findings of the association for computational linguistics: NAACL 2025*, 295–312.

[3] Zehui Wang, Wolfram Höpken, and Dietmar Jannach. 2025. Beyond visit trajectories: enhancing POI recommendation via LLM-augmented text and image representations. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, 521–526.

[4] Min Hou, Chenxi Bai, Le Wu, Hao Liu, Kai Zhang, Weiwen Liu, Richang Hong, Ruiming Tang, and Meng Wang. 2026. RecCocktail: a generalizable and efficient framework for LLM-based recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence* number 17. Vol. 40, 14838–14847.

[5] Jacob Munson, Thomas Cuezze, Siddat Nesar, and Dominique Zosso. 2025. A review of large language models and the recommendation task. *Discover Artificial Intelligence*, 5, 1, 203.

[6] Hao Liao, Jiwei Zhang, Jianxun Lian, Wensheng Lu, Mingqi Wu, Yong Zhang, Yitian Huang, Mingyang Zhou, Rui Mao, et al. 2026. Eliminating out-of-domain recommendations in LLM-based recommender systems: a unified view. In *Findings of the Association for Computational Linguistics: ACL 2026*, 6251–6271.

[7] Xinyu Li, Chuang Zhao, Hongke Zhao, Likang Wu, Ming He, and Jianping Fan. 2025. GANPrompt: improving LLM-based recommendations with GAN-enhanced diversity prompts. *ACM Transactions on Information Systems*, 44, 1, 1–36.

[8] Jizhi Zhang, Keqin Bao, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. Is ChatGPT fair for recommendation? evaluating fairness in large language model recommendation. In *Proceedings of the 17th ACM conference on recommender systems*, 993–999.

[9] Chandan Kumar Sah, Xiaoli Lian, Tony Xu, and Li Zhang. 2025. FairEval: evaluating fairness in LLM-based recommendations with personality awareness. *arXiv preprint arXiv:2504.07801*.

[10] Yao Wu, Jian Cao, and Guandong Xu. 2023. Fairness in recommender systems: evaluation approaches and assurance strategies. *ACM Transactions on Knowledge Discovery from Data*, 18, 1, 1–37.

[11] Bjørnar Vassøy and Helge Langseth. 2024. Consumer-side fairness in recommender systems: a systematic survey of methods and evaluation. *Artificial Intelligence Review*, 57, 4, 101.

[12] Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. TALLRec: an effective and efficient tuning framework to align large language model with recommendation. In *Proceedings of the 17th ACM conference on recommender systems*, 1007–1014.

[13] Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang, and Xiangnan He. 2024. LLaRA: large language-recommendation assistant. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, 1785–1795.

[14] Jianling Wang, Haokai Lu, James Caverlee, Ed H Chi, and Minmin Chen. 2024. Large language models as data augmenters for cold-start item recommendation. In *Companion proceedings of the ACM web conference 2024*, 726–729.

[15] Jianghao Lin et al. 2025. How can recommender systems benefit from large language models: a survey. *ACM Transactions on Information Systems*, 43, 2, 1–47.

[16] Wenyue Hua, Yingqiang Ge, Shuyuan Xu, Jianchao Ji, Zelong Li, and Yongfeng Zhang. 2024. UP5: unbiased foundation model for fairness-aware recommendation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1899–1912.

[17] Arya Fayyazi, Mehdi Kamal, and Massoud Pedram. 2025. FACTER: fairness-aware conformal thresholding and prompt engineering for enabling fair LLM-based recommender systems. *arXiv preprint arXiv:2502.02966*.

[18] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as language processing (RLP): a unified pretrain, personalized prompt & predict paradigm (P5). *arXiv preprint arXiv:2203.13366*.

[19] Shuyuan Xu, Wenyue Hua, and Yongfeng Zhang. 2024. OpenP5: an open-source platform for developing, training, and evaluating LLM-based recommender systems. *arXiv preprint arXiv:2306.11134*.

[20] Aleksandr V. Petrov and Craig Macdonald. 2023. Generative sequential recommendation with GPTRec. *arXiv preprint arXiv:2306.11114*.

[21] Panfeng Cao and Pietro Lio. 2024. GenRec: generative sequential recommendation with large language models. *arXiv preprint arXiv:2407.21191*.

[22] Jian Xu, Sichun Luo, Xiangyu Chen, Haoming Huang, Hanxu Hou, and Linqi Song. 2025. RALLRec: improving retrieval augmented large language model recommendation with representation learning. In *Companion Proceedings of the ACM on Web Conference 2025*, 1436–1440.

[23] Wonbin Kweon, Sanghwan Jang, SeongKu Kang, and Hwanjo Yu. 2025. Uncertainty quantification and decomposition for LLM-based recommendation. In *Proceedings of the ACM Web Conference 2025*, 4889–4901. doi:10.1145/3696410.3714601.

[24] Alexandre Andre, Gauthier Roy, Eva Dyer, and Kai Wang. 2025. Revealing potential biases in LLM-based recommender systems in the cold start setting. *arXiv preprint arXiv:2508.20401*.

[25] Lei Chen, Le Wu, Kun Zhang, Richang Hong, Defu Lian, Zhiqiang Zhang, Jun Zhou, and Meng Wang. 2023. Improving recommendation fairness via data augmentation. In *Proceedings of the ACM web conference 2023*, 1012–1020.

[26] Alessandro B Melchiorre, Navid Rekabsaz, Emilia Parada-Cabaleiro, Stefan Brandl, Oleg Lesota, and Markus Schedl. 2021. Investigating gender fairness of recommendation algorithms in the music domain. *Information Processing & Management*, 58, 5, 102666.

[27] Xinyu Lin, Wenjie Wang, Yongqi Li, Shuo Yang, Fuli Feng, Yinwei Wei, and Tat-Seng Chua. 2024. Data-efficient fine-tuning for LLM-based recommendation. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, 365–374.

[28] Dugang Liu, Shenxian Xian, Xiaolin Lin, Xiaolian Zhang, Hong Zhu, Yuan Fang, Zhen Chen, and Zhong Ming. 2024. A practice-friendly LLM-enhanced paradigm with preference parsing for sequential recommendation. *arXiv preprint arXiv:2406.00333*.

[29] Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. 2024. State of what art? a call for multi-prompt LLM evaluation. *Transactions of the Association for Computational Linguistics*, 12, 933–949. doi:10.1162/tacl_a_00681.

[30] Anwoy Chatterjee, H S V N S Kowndinya Renduchintala, Sumit Bhatia, and Tanmoy Chakraborty. 2024. POSIX: a prompt sensitivity index for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, (Eds.) Association for Computational Linguistics, Miami, Florida, USA, (Nov. 2024), 14550–14565. doi:10.18653/v1/2024.findings-emnlp.852.

[31] Yao Qiang, Subhrangshu Nandi, Ninareh Mehrabi, Greg Ver Steeg, Anoop Kumar, Anna Rumshisky, and Aram Galstyan. 2024. Prompt perturbation consistency learning for robust language models. In *Findings of the Association for Computational Linguistics: EACL 2024*. Yvette Graham and Matthew Purver, (Eds.) Association for Computational Linguistics, St. Julian's, Malta, (Mar. 2024), 1357–1370. doi:10.18653/v1/2024.findings-eacl.91.

[32] F Maxwell Harper and Joseph A Konstan. 2015. The MovieLens datasets: history and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 5, 4, 1–19.

[33] Aaron Grattafiori et al. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

[34] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*. Vol. 36.

[35] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations*. OpenReview.net. https://openreview.net/forum?id=nZeVKeeFYf9.

[36] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, 611–626. doi:10.1145/3600006.3613165.

[37] Jingzhe Liu, Hanbing Wang, Jiliang Tang, Liam Collins, Tong Zhao, Neil Shah, and Mingxuan Ju. 2026. Beyond fixed depths and widths: optimizing textual decoding tries in LLM-based generative recommendation. *arXiv preprint arXiv:2607.16633*.