A Description is Worth a Million Clicks

André F. Cruz[1,2] Andreas Bjerre-Nielsen[3,4,5] Jon Kleinberg[6] Rediet Abebe[1,2,7]

[1] *Max Planck Institute for Intelligent Systems, Tübingen* [2] *Tübingen AI Center*
[3] *University of Copenhagen* [4] *CAISA* [5] *Statistics Denmark*
[6] *Cornell University* [7] *ELLIS Institute Tübingen*

Abstract

Representations of human preferences guide search, content feeds, advertising, and large language model (LLM) alignment. For decades, these representations have been learned from fixed-form signals (ratings, likes, clicks), and mature, scalable algorithms such as collaborative filtering were built around them. Yet people express their preferences far more naturally in language. LLMs now make it possible to work with such text directly. On six large-scale datasets from Amazon and Goodreads, we compare collaborative filtering, trained and heavily tuned on abundant interaction data, against a zero-shot LLM prompted with item descriptions and users' past reviews. This simple text baseline is competitive with collaborative filtering trained on around *one million* interactions. To probe why, we use LLMs to distill text into thousands of interpretable item characteristics and each user's stated preferences over them. These features predict choices just as well as the text they come from. This is consistent with Lancaster's view of goods as bundles of characteristics: a review speaks to characteristics shared by many items, whereas an interaction records a single choice. Fixed-form and free-form representations are also complementary, jointly outperforming any single model. Together, these findings invite revisiting how preferences are elicited and encoded.

1 Introduction

Human preferences play a central role in many decision-making systems, from allocating public resources and matching students to schools, to recommending books, movies, and restaurants (1–3). In all such settings, preferences are usually translated into a structured format that can be stored, compared, and optimized (4, 5). While these encodings are often treated as neutral inputs, they rarely reflect how people naturally articulate their preferences (1, 6). Instead, commonly used representations are shaped by what can be easily elicited at scale and by the algorithms available to process them (5, 7, 8). In practice, both pressures favor *fixed-form* signals: star ratings, binary likes, or implicit interaction traces such as clicks and views (9, 10). These signals are abundant and admit scalable, well-studied algorithms such as collaborative filtering (9, 11–13). However, fixed-form signals are intrinsically lossy: they encode *what* a user chose, but not *why* (14–17); they are biased by platform exposure (18); and they degrade sharply in cold-start settings, where historical data is sparse (19). More fundamentally, preferences expressed over individual items can fail to recover preferences over whole sets, such as for fairness or variety (5, 20).

Correspondence to André F. Cruz: andre.cruz@tuebingen.mpg.de

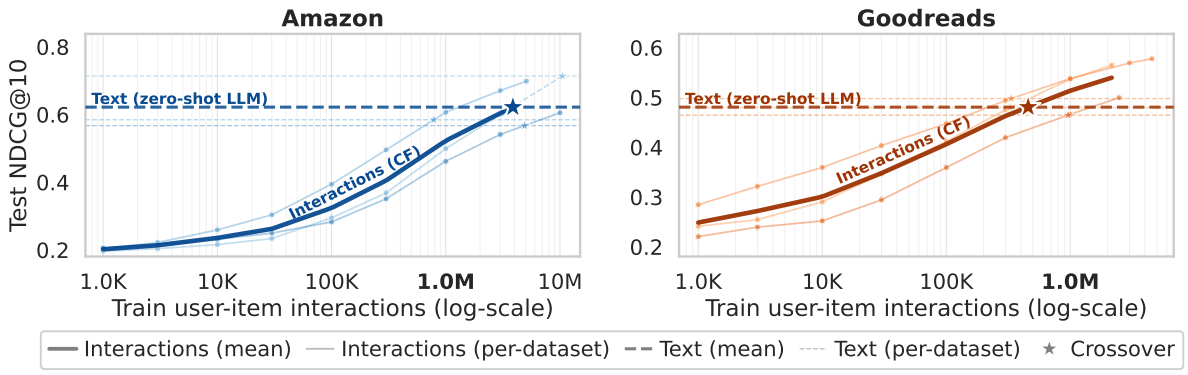


Figure 1: **Interactions-based preference model needs around 1M training interactions to match an off-the-shelf Text-based model given a user's past reviews and item descriptions.** Test-set recommendation quality (NDCG@10) of matrix factorization collaborative filtering (CF) as training data grows, on Amazon and Goodreads (3 datasets each). Dashed line: Text-based zero-shot GPT-5 using user reviews. Thicker lines show (model, domain) averages, thin lines show per-dataset results. Crossover (★) at 0.8–10M on Amazon and 0.3–1M on Goodreads. Each point: best of 154 CF hyperparameter tuning trials by validation NDCG@10 (8,316 total).

By contrast, users routinely articulate preferences directly in natural language. Unlike clicks or views, which reveal preferences only indirectly, through choices of isolated items (10), text lets users state what makes items valuable: attributes, trade-offs, and contextual explanations (21, 22). Yet representation choices in ML have long been driven by what models could ingest. Early ML researchers classified Iris flowers from petal and sepal measurements (23). A table of four numbers hardly described a flower better than a photograph, but models at the time could not efficiently process images. Fixed-form interactions occupy a similar role: easy to process, but poor descriptions of what users want. Engagement signals such as clicks and views can also diverge from users' broader goals, and engagement-optimizing systems can exploit this gap (24, 25). Text instead records what users say they want, which may track their goals more closely (25).

Historically, free-form text has been difficult to operationalize at scale. Earlier recommender systems incorporated language primarily through bag-of-words features, topic models, or features hand-engineered by domain experts (14, 26, 27). These approaches were brittle and task-specific, limiting the generalizable preference signal that could be extracted from text. Even after neural text models were adopted (28), text-only approaches generally did not outperform well-tuned interaction-based methods (29, 30). As a result, text was typically found most useful as an aid in cold-start settings, and contributed little as platforms acquired more interaction data (14, 30).

Recent advances in large language models (LLMs) materially change this landscape. Modern LLMs can abstract and categorize user-authored text at scale, producing semantically meaningful representations across domains (31, 32). Persistent "memory" features in LLM assistants provide a familiar example: they record users' preferences and context to personalize future responses (33). Mechanism design and social choice are drawing on the same capability, using LLMs to elicit preferences in natural language for course allocation and collective decision-making (34, 35). In light of this, we revisit a longstanding preference elicitation question: when text is available, how informative is it relative to fixed-form interaction data? Appendix A reviews related work in depth.

2 Overview

This paper studies natural language as a first-class source of preference information. Crucially, we acknowledge a persistent asymmetry between elicitation formats: while text is inherently more expressive, interaction data is generated passively at scale as a byproduct of system use, making it far more plentiful. This asymmetry motivates our central question: how much task-specific training interaction data must collaborative filtering accumulate before it matches a simple, zero-shot, off-the-shelf LLM given each user's own text? Even if only predictive performance matters, the answer bounds when a new platform can rely on language alone and when an incumbent's interaction history begins to pay off. Beyond performance, users can freely inspect, correct, or withhold the preferences they state in language, but not those revealed through clicks in a constrained interface. We preserve this asymmetry in all our experiments and evaluation: interaction-based models train on abundant historical data, and the off-the-shelf language models behind our text-based representations receive no task-specific fine-tuning.

We construct three parallel representations of user preferences from the same underlying data: (i) user–item interaction vectors; (ii) free-form preference descriptions distilled from user-authored reviews using an off-the-shelf LLM; and (iii) interpretable categorical features extracted from the same reviews with an LLM (preferences over attributes rather than items). We evaluate these representations on six large public datasets spanning two platforms: three book genres from `Goodreads.com` (36) and three product categories from `Amazon.com` (37).

Our contributions can be summarized by three main findings:

- **Free-form text is a competitive alternative to abundant interaction data.** Zero-shot text representations outperform interaction data (with collaborative filtering) until it reaches approximately one million training interactions (Figure 1).
- **LLMs distill text into item characteristics and users' stated preferences over them.** Scored by a small supervised model, these categorical features match the average performance of the free-form text representation.
- **Preference representations carry complementary signal.** Interaction data, free-form text, and categorical features encode distinct aspects of user preferences; simple score-averaging ensembles achieve the strongest overall performance.

3 Experimental setup

This section describes the data, preference representations, and evaluation protocol behind these findings. We structure the setup so that (i) all representations are derived from the same users, items, and observed behaviors; (ii) each is drawn from organic real-world data, which preserves naturally occurring differences in scale, coverage, and quality; and (iii) modeling choices are kept simple and standard. Our central comparison, text versus interactions, varies the size of the collaborative-filtering training set, and asks "How many train interactions must a tuned recommender ingest to match a zero-shot text predictor?"

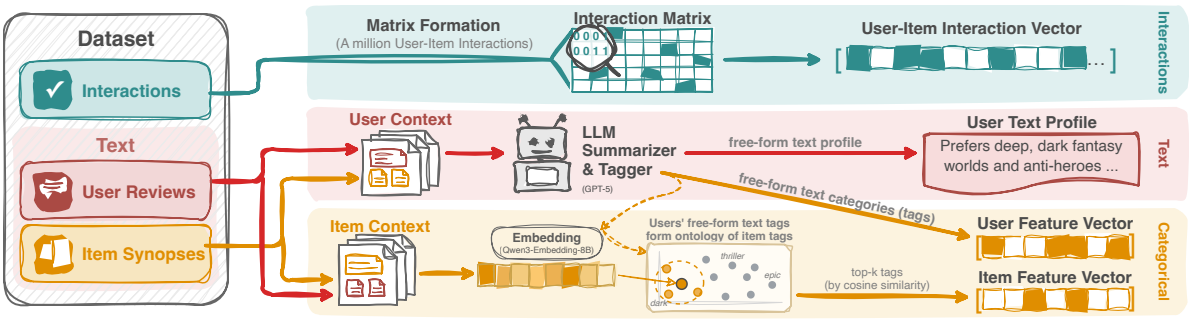


Figure 2: **Deriving three parallel preference representations from a shared dataset.** [*Left*] Each dataset provides user–item interactions, user-authored reviews, and text descriptions of each item. [*Top*, **cyan**] Interactions: a sparse binary user–item matrix $\mathbf{M}$ trains collaborative-filtering models. [*Middle*, **red**] Text: an off-the-shelf LLM summarizes a user's past reviews and the corresponding item descriptions into a free-form text profile, which an LLM later reads to score candidate items. [*Bottom*, **yellow**] Categorical: from the same user context, the LLM also emits short free-form preference tags (e.g., *dark*, *epic*, *thriller*) with no predefined ontology; the union of users' tags defines the categorical vocabulary. Each item is placed in the same vocabulary by embedding its description and up to five reviews, and assigning its 5 nearest tags by cosine similarity. Users and items become interpretable vectors over a shared space of preference characteristics.

3.1 Data

We draw on two large-scale public collections of user reviews and interactions: `Goodreads.com` book reviews (36), and `Amazon.com` product reviews (37). Both include information on each item (book or product), user–item interactions (reads or purchases), and free-form reviews written by users, which form the basis for our preference representations (Figure 2). Goodreads is the world's largest book cataloging platform, where users track, rate, and review books they have read; an *interaction* corresponds to a book that was both read and rated by a user (for consistency with Amazon, whose data include only rated purchases). Amazon is the world's largest online retailer; an Amazon *interaction* corresponds to a product that was both purchased and rated by a user. From each platform we select three subsets, yielding six datasets in total: the Goodreads genres "Fantasy & Paranormal", "Mystery, Thriller & Crime", and "History & Biography", and the Amazon product categories "Digital Music", "Movies & TV", and "Toys & Games". Appendix C details data cleaning, preprocessing, and filtering statistics.

3.2 Preference representations

Figure 2 shows how we derive parallel representations from three sources of preference information: (i) user–item interactions, (ii) user-written reviews, and (iii) item descriptions.

Interaction matrix. The standard interaction representation is a sparse user–item matrix $\mathbf{M}$ whose entry $M_{u,i}$ indicates whether user u interacted with item i (e.g., user u bought and rated product i). This formulation underlies a large body of recommender-system research and widespread usage in production systems (9, 11, 12, 38). However, it has clear drawbacks. A user's row in $\mathbf{M}$ is a bare log of past actions: preferences cannot be read off it directly, and the latent factors that models fit to it are opaque, so users cannot inspect or correct how the system represents them, nor

choose which of their actions inform it. The signal is also indirect: each entry records a reaction, shaped by what the platform exposed and what was popular (18), rather than the intent behind the choice (15). Finally, **M** needs many interactions to fit well and is unreliable when they are scarce.

Text preference descriptions. User-written reviews provide richer and more interpretable signals than interaction logs: they explain why a user liked or disliked a title and expose recurring themes that may generalize across books. We use pretrained LLMs as zero-shot summarizers to produce a concise, interpretable description of each user's reading preferences. For each user, we construct a "preference context" that aggregates text from their reading history. It contains the book synopsis and the user's review for every reviewed book, plus frequency statistics over the user's most-read authors and sub-genres (computed over the same set of reviewed books), which make aggregate patterns explicit to the summarizer. The model generates a compact textual profile, limited to at most three paragraphs, describing the user's general preferences. Appendix F shows the exact prompt templates and example profile instances. The length constraint imposes an information bottleneck that favors stable, cross-title themes over item-specific detail. Summaries of users with many reviews tend to be more accurate and generalizable; those from sparse profiles are noisier. We treat these summaries as a standalone, human-readable representation of user preferences that can be inspected and corrected.

This personalized text profile mirrors the user-controllable "memory" features now common in LLM assistants (e.g., ChatGPT), which keep a persistent, natural-language record of user preferences, used to tailor responses to each user. This representation has drawbacks of its own. A textual profile requires either deliberate input from the user or organic user-authored text, such as the reviews we use here (Appendix F.2). Cost is a further drawback: unlike interaction-based models, the text representation requires LLM inference (Appendix E.5). The summarizing LLM may also introduce errors or biases.

Categorical features. In addition to free-form text summaries, we derive a structured, categorical representation of user preferences from LLM-generated tags. For each user, the LLM produces a set of short preference descriptors (e.g., `epic-fantasy`, `female-lead`, `complex-world-building`) intended to capture salient aspects of the user's preferences. We impose no predefined ontology on these tags, so the model can surface distinctions it deems important. The union of all users' tags defines the categorical vocabulary. Each user is then represented by a vector over this vocabulary that indicates which tags they were assigned, yielding an interpretable categorical feature representation derived from their textual reviews.

To place items in the same space, we use a text embedding model (`Qwen3-Embedding-8B` (32)): we embed every tag in the vocabulary, embed each item's description together with up to five randomly sampled community reviews, and assign each item its 5 nearest tags by cosine similarity (Prompt F.9). Users and items thus become feature vectors over a shared categorical space of user preferences and item attributes. In terms of Lancaster's classic model of consumer choice (21), item tagging estimates the *consumption technology* that maps goods to their characteristics, which applied work has typically supplied by hand (Appendix B.1). For users, the tags record which of these characteristics each user values, as expressed in their own reviews rather than inferred from their choices. Because free-form tags can overlap semantically (e.g., `fast-paced` and `high-octane`), we also tested consolidating the vocabulary by clustering tag embeddings; this did not improve validation performance, so all reported results use the full tag vocabulary (Appendix D).

3.3 Models

This section describes how each representation is mapped to a score that ranks items for a user. Our text and categorical models use deliberately simple choices (e.g., zero-shot, untuned prompts) that could be improved, so our results present a lower bound on what users' text can achieve.

Interaction-based models. We include two models that operate directly on user–item interaction data: collaborative filtering and a popularity baseline. An interaction is binary: for user u and item i, it is 1 if the user has read/bought and rated the item, and 0 otherwise. Because training-set size is our central comparison axis, we ablate the interaction-based models across training-set sizes by sub-sampling the train interactions, from small subsets up to the full 1M train interactions (Table C.1). All performance results are reported on held-out interactions.

Collaborative filtering. We use a collaborative filtering (CF) model based on matrix factorization for implicit feedback, which remains the standard approach for modeling large-scale interaction data (9, 39). Let $\mathbf{M} \in \mathbb{R}^{n \times m}$ denote the resulting binary user–item interaction matrix. The model learns latent user and item factors, $\mathbf{U} \in \mathbb{R}^{n \times d}$ and $\mathbf{V} \in \mathbb{R}^{m \times d}$, such that $\mathbf{M} \approx \mathbf{U}\mathbf{V}^\top$. User–item affinity is the inner product $\langle \mathbf{U}_u, \mathbf{V}_i \rangle$. We implement this model using the open-source `implicit` library (40). At evaluation time, we *fold in* each validation/test user by holding the item factors $\mathbf{V}$ fixed (learned from train interactions) and solving a regularized least-squares problem to infer that user's latent vector $\mathbf{U}_u$ from their profile interactions (9). Folding-in is an evaluation-time step, distinct from the training-set-size axis, which varies how much train data the item factors $\mathbf{V}$ are learned from.

Popularity. The popularity baseline model ranks items by their total number of interactions in the training data. This model produces the same ranking for all users. That is, at evaluation time, for each user, it sorts the user's 50 candidate items by relative popularity.

Text-based model. Our text-based model uses only natural language inputs and zero-shot prompting of a large language model. We use GPT-5 (41) for test users' profiles, tags, and scores. To reduce cost, validation users' profiles, tags, and scores come from GPT-5-mini. Appendix E.6 repeats the test pipeline on one dataset with five other LLMs. For each user–item pair, the prompt includes (i) a task description, (ii) the user's text profile (LLM-generated), (iii) the item description, and (iv) up to five top-voted community reviews of the item (from the train set). The model is prompted to output a score from 0 to 10 representing how likely the user is to engage with the item (e.g., to buy the item on Amazon). Appendix F gives prompt templates and scoring details.

Categorical model. For categorical representations extracted from text, users and items are fixed-length feature vectors. To score a user–item pair (u, i), a small supervised model (a multilayer perceptron, MLP) maps the user's and item's categorical feature vectors to a relevance score (Appendix D). Training has two phases: we first tune hyperparameters on a validation-only fit, then refit the best configuration for the test evaluation on all 2,000 evaluation users (the profile interactions of validation and test users). Held-out interactions are naturally unused. Hyperparameter tuning does not meaningfully improve the MLP over its default configuration: across datasets, tuning changes test NDCG@10 by at most 0.018 (mean +0.010), roughly an order of magnitude less than the +0.11 to +0.16 gains that tuning brings collaborative filtering (Appendix E.7, Table E.8).

3.4 Evaluation

We evaluate how well each preference representation predicts held-out interactions in a shared ranking task. Evaluation users must provide enough history to build a representation while retaining

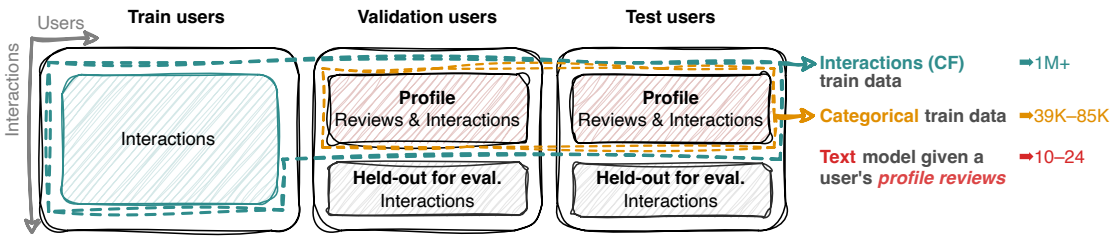


Figure 3: **The comparison is deliberately asymmetric: Interactions (CF) trains on 1M+ interactions, while Text is zero-shot over a user's 10–24 profile reviews.** We split users into *train*, *validation*, and *test* sets (columns). Each validation and test user's interactions divide into a *profile* set, which builds that user's representation, and a *held-out* set of 10 interactions for scoring. Interactions model (CF) trains on the full 1M-interaction training set and folds in each evaluation user from their profile interactions to fit that user's latent factors. Categorical model (MLP) is fed the <user, item> categorical representations of all evaluation users' profile interactions (39K–85K interactions), plus randomly sampled negative examples. Text model builds a text profile from only the target user's profile reviews (Table C.3) and the corresponding item descriptions.

held-out interactions for testing. Figure 3 summarizes the split between train/validation/test users, and each user's split into profile (usable for training) and held-out (for evaluation) interactions.

Split in train, validation, and test. We split at the user level. Validation and test each contain 1,000 users sampled uniformly from users who (i) interacted with at least 15 items and (ii) wrote at least 5 reviews. Condition (i) ensures each user has enough held-out positives; condition (ii) ensures text and categorical representations can be derived from user-authored language. All other users form the training set. Table C.3 reports split statistics per dataset.

Profile vs. evaluation interactions. Because models require user-specific information, we split each validation/test user's interactions into (a) a *profile* set and (b) an *evaluation* set. Not every interaction carries a written review. We first sample each user's evaluation set: 10 interactions selected at random, prioritizing interactions without a review so that review text is preserved for the profile. The profile set then consists of the remaining interactions that do have a review; these profile reviews are the only user-specific textual signal used to construct the user's text profile and categorical features. We denote by k the number of profile interactions (equivalently, the number of profile reviews) available for a user at evaluation time, its *profile length*; k varies across users, allowing evaluation from cold-start (small k) to information-rich regimes (large k). As previously mentioned, profile interactions are also used by the collaborative filtering model to fit user-specific latent factors at inference time (*fold in*). All models therefore receive the same *number* of inference-time interactions (k) for a given evaluation user, and differ only in the type of input data (text, categories, interactions vector) and how they use them (Figure 3).

Candidate set and metric. We evaluate all models on the same candidate set for each user. For a given evaluation user, we form 50 candidate items: 10 held-out positives from the evaluation set and 40 negatives sampled from items the user did not interact with, with sampling weighted by item popularity to preserve overall item-prevalence effects. Each model scores and ranks the 50 candidates. NDCG@10 is our primary metric: all tuned models (collaborative filtering and the categorical MLP) select hyperparameters by validation NDCG@10. Normalized discounted

cumulative gain at rank 10 (NDCG@10) rewards rankings that place held-out positives near the top, with a logarithmic discount by rank (42). For a single user,

$$\text{NDCG@10} = \frac{1}{Z} \sum_{j=1}^{10} \frac{\text{rel}_j}{\log_2(j+1)},$$

where $\text{rel}_j \in \{0, 1\}$ indicates whether the item at rank j is a held-out positive, and $Z = \sum_{j=1}^{10} 1/\log_2(j+1)$ normalizes by the ideal ranking, so NDCG@10 equals 1 exactly when the 10 positives fill the top 10 ranks. We report three further ranking-quality metrics: MRR, Recall@10, and ROC AUC. All metrics are computed per user and averaged across users.

Leakage and memorization. Both platforms' data are public and predate the LLM's training, so a natural concern is that the text model predicts held-out interactions by recalling memorized data rather than by reading each user's profile. Direct leakage is excluded by construction: no prompt contains held-out interactions, and all community reviews are drawn from the training split, never from the evaluation user. Memorized pretraining data is harder to rule out, but the scoring model never observes the user's own reviews, only the three-paragraph user profile, and two experimental results further argue against memorization (Appendix F.1). The popularity baseline, which captures the most easily memorized signal, trails the text model by a wide margin on every dataset (Table 1), and the categorical model matches the text model's performance while reducing each user to an average of 17 human-readable preference tags.

4 Results

We compare three user preference representations: interaction-based collaborative filtering, text-based user profiles read by an LLM, and discrete categorical features extracted from the same text. We study text versus interactions as a function of the number of interactions on which CF is trained, since the central question is how much task-specific interaction data CF needs to match a zero-shot text predictor. Table 1 reports user-wise test performance for the base models across the six datasets and four ranking metrics (NDCG@10, MRR, AUC, Recall@10). Appendix E has more results.

4.1 Text vs. user–item interactions

Figure 4 compares a text-based predictor (zero-shot GPT-5) to collaborative filtering (CF) as a function of the number of interactions CF is trained on. Across all six datasets, what users express in text provides a reliable preference signal. For a given evaluation user, both models receive the same number of profile interactions (k); the only quantity varied along the x-axis is the number of *train* interactions used to fit CF, isolating the effect of CF's training scale. CF matches text only once trained on roughly one million interactions (0.3–1M on Goodreads, 0.8–10M on Amazon), despite extensive hyperparameter tuning (Figure 1). More training interactions yield better item factors and denser coverage of the catalog, so CF improves predictably with training scale. By contrast, the text-based predictor is a fixed, zero-shot LLM; it is not fitted to the interaction data, yet it remains competitive until CF reaches the million-interaction scale. The crossover is not specific to GPT-5. On AZ Movies&TV, we repeated the comparison with five other LLMs from OpenAI and Anthropic, spanning roughly two orders of magnitude in API cost; CF matches each of them after 0.3–4M training interactions, depending on the LLM and metric (Figure E.6, Appendix E.6). The crossover comes latest for the strongest zero-shot scorer, Claude Opus 5.5 (2–4M).

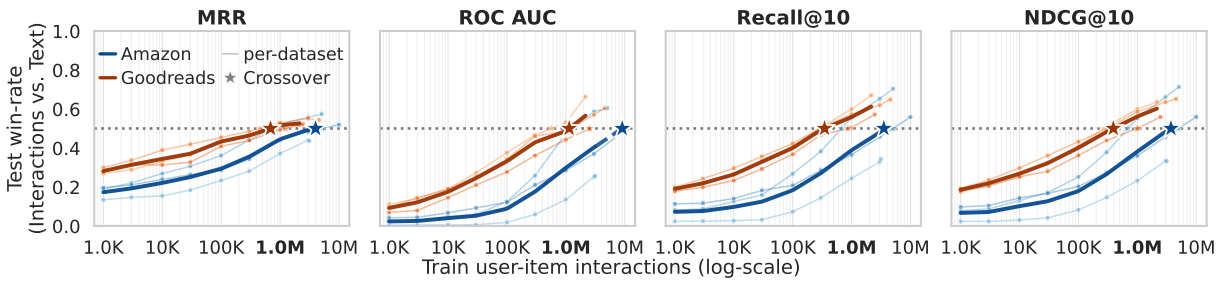


Figure 4: **Across four ranking metrics, collaborative filtering overtakes the zero-shot text model only after roughly one million training interactions.** Per-user win-rate of Interactions (CF) over Text (zero-shot GPT-5): for each panel's metric, the fraction of test users for which CF scores higher than Text, as CF's training set grows (x-axis, log scale). The dotted line marks parity (50% win-rate); stars mark the per-domain crossover. Thick lines are domain averages (Amazon, Goodreads); thin lines are individual datasets. Each CF point is the best of 154 tuning trials.

The same result holds across user regimes: text is competitive for both cold-start and warm-start users. Appendix E.2, Figures E.1–E.4, splits test users into cold-start ($k < 10$ profile reviews) and warm-start ($k > 40$) cohorts, across all four ranking metrics. Higher k makes prediction harder, since users with longer histories tend to interact with less popular items, leading the popularity baseline to degrade sharply for data-rich users who drift toward long-tail items. Figure C.1 (Appendix C.6) shows mean item popularity as a function of profile length k. Profile length therefore has two opposing effects: longer profiles carry a stronger user-specific preference signal, while the accompanying drift toward less popular items makes the prediction task harder. On our datasets, the two roughly balance out, and model performance is stable across the cold- to warm-start range.

4.2 Categorical features recover interpretable characteristics

Text profiles are human-legible, but using an LLM to score each user–item pair is expensive: in our experiments, GPT-5 scoring costs 1.45 cents per pair on average (95% CI: 1.25–1.65 cents; Appendix E.5, Table E.4). Profile generation costs 4.33 cents per user (95% CI: 3.53–5.13 cents), but it runs once per user rather than once per candidate item, so scoring accounts for over 90% of the total cost. Scoring is quickly getting cheaper: with GPT-6-luna, a recent, smaller, and more efficient model, it costs 0.02 cents per pair on AZ Movies&TV and total API cost falls by 98%, while zero-shot AUC and NDCG@10 drop by only 0.02 (Appendix E.6). Categorical features derived from the same text provide a more efficient alternative, while remaining interpretable. To test whether text's predictive signal can be recovered in a structured, interpretable form, we derive discrete categorical features from the same user-authored reviews. A small supervised MLP then maps the user and item categorical feature vectors to a relevance score; we fit it on 39K–85K interactions per dataset (the profile interactions of the 2,000 evaluation users), light supervision relative to the 1M interactions used to train collaborative filtering. The MLP is itself not interpretable, but each user's and item's representation is. Where full end-to-end interpretability is required, the MLP can be replaced by a simpler scoring rule over the same features, at the cost of some performance.

Figure E.5 (Appendix E.3) compares the categorical model to the text-based LLM across profile lengths. Across datasets and user cohorts, the categorical model roughly matches the text-based LLM. The model also rivals or outperforms collaborative filtering, which infers the same users' tastes

Table 1: Per-user test-set performance of the base preference models on six datasets from Goodreads (*GR*) and Amazon (*AZ*). Bold: best per (dataset, metric) and models tied with it (paired Wilcoxon test, $p > 0.05$). The Text block reports two models built from the same reviews: a zero-shot LLM scoring free-form text, and a supervised MLP over categorical features extracted from that text. The Interactions (CF) block reports four settings: 1M or 100K training interactions, each tuned on validation NDCG@10 or left at library defaults. Ensembles appear in Figure 5 and Table E.1.

Preference representation (model)		AZ Toys&Games				AZ Movies&TV				AZ Music			
		NDCG	MRR	AUC	R@10	NDCG	MRR	AUC	R@10	NDCG	MRR	AUC	R@10
Text	*free-form (ZS LLM)*	0.567	**0.818**	0.771	0.522	0.584	0.822	0.800	0.540	**0.713**	**0.922**	**0.880**	**0.662**
	categorical (MLP)	**0.595**	**0.816**	**0.803**	**0.558**	**0.679**	**0.902**	**0.871**	**0.630**	0.699	0.899	**0.885**	**0.654**
Interactions (CF)	*1M, tuned*	0.461	0.746	0.678	0.419	0.606	0.858	0.785	0.560	0.498	0.762	0.709	0.453
	1M, default	0.299	0.529	0.586	0.284	0.458	0.704	0.684	0.434	0.349	0.576	0.611	0.332
	100K, tuned	0.282	0.524	0.546	0.265	0.394	0.648	0.611	0.365	0.294	0.559	0.555	0.271
	100K, default	0.261	0.509	0.517	0.244	0.344	0.607	0.544	0.325	0.262	0.524	0.518	0.245
Popularity		0.233	0.422	0.559	0.231	0.330	0.494	0.707	0.344	0.234	0.394	0.550	0.242
Preference representation (model)		GR History				GR Fantasy				GR Thriller			
		NDCG	MRR	AUC	R@10	NDCG	MRR	AUC	R@10	NDCG	MRR	AUC	R@10
Text	*free-form (ZS LLM)*	0.464	**0.742**	0.734	0.433	0.497	0.778	0.747	0.455	0.479	0.763	0.720	0.436
	categorical (MLP)	**0.505**	**0.759**	**0.773**	**0.473**	**0.531**	0.777	**0.793**	**0.497**	**0.535**	**0.786**	**0.779**	**0.497**
Interactions (CF)	*1M, tuned*	0.466	0.725	0.708	0.432	**0.538**	**0.814**	0.741	**0.489**	**0.536**	**0.796**	0.735	**0.489**
	1M, default	0.355	0.612	0.633	0.332	0.392	0.673	0.642	0.353	0.382	0.654	0.657	0.348
	100K, tuned	0.359	0.613	0.631	0.337	0.447	0.731	0.679	0.405	0.411	0.682	0.660	0.374
	100K, default	0.304	0.544	0.586	0.289	0.376	0.671	0.619	0.335	0.331	0.604	0.603	0.303
Popularity		0.180	0.342	0.497	0.183	0.236	0.413	0.541	0.236	0.206	0.384	0.489	0.204

from their choices by training on more than an order of magnitude more interactions (Table 1). Both the text and categorical models draw on the same reviews, but the LLM reads the full text profile and the categorical model sees only the explicit tags. Their similar performance does not necessarily imply that the LLM relies on such categories internally, but it does show that recurring, human-interpretable categories can capture the predictive signal in free-form text. Neither model dominates and each scores higher on some (dataset, metric) pairs (Table 1). The categorical model matches the LLM despite a much coarser representation of the same reviews, likely because it is trained on task-specific data while the LLM predicts zero-shot. This suggests headroom in the text representation itself: a scorer trained on the same data could extract more from the full profiles than zero-shot scoring does. Stronger LLMs mainly benefit the zero-shot scorer: across six LLMs on AZ Movies&TV, the LLM scorer NDCG@10 varies by 0.13, against only 0.02 for the categorical model built on the same LLM's tags (Appendix E.6).

The categorical model's training data include the test users' own profile interactions, which collaborative filtering also uses to fit each test user's latent factors. Fit on the 1,000 validation users alone, the MLP loses 0.06–0.07 NDCG@10 on Amazon and 0.10–0.17 on Goodreads, and falls below the text model on five of six datasets (Appendix E.4). Its match with the text model therefore holds for a model fit on the available profile interactions, as a deployed model could be. The larger Goodreads loss is consistent with sparse tag coverage: more than half of Goodreads test users' tag occurrences involve tags that no validation user carries, against about a third on Amazon. We tested this by grouping tags into 100 clusters, all of which occur among validation users, so every test tag maps to a feature seen in training. This shrinks the Goodreads loss to 0.04–0.06, close to Amazon's 0.02–0.05 under the same clustering, which supports the coverage explanation (Table E.3).

4.3 Complementary preference signals

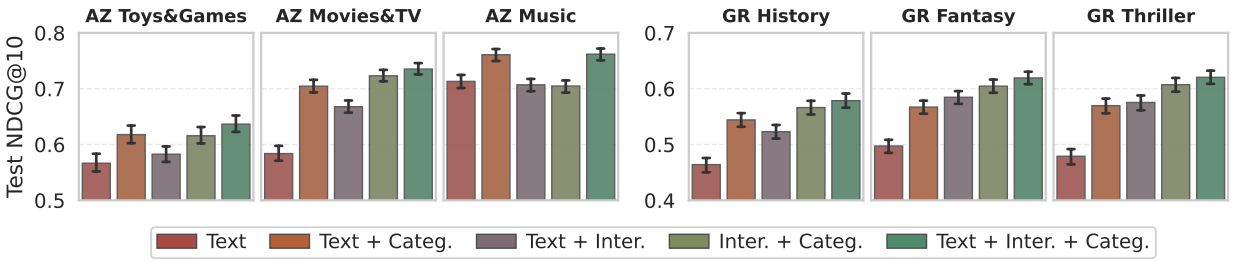


Figure 5: **The three-way ensemble is best or tied-for-best on every dataset.** Test NDCG@10 of the text model and four ensembles, averaged over test users; error bars are 95% bootstrap confidence intervals. MRR, AUC, and Recall@10 show the same pattern (Table E.1).

Having shown that text and categorical representations each carry strong signal, we test whether they capture complementary information. We construct simple ensembles by averaging predicted relevance scores across parallel user representations. Figure 5 shows the pairwise and three-way ensembles at NDCG@10; Table E.1 compares base models and ensembles on all four metrics.

Ensembling text-based and categorical representations yields consistent performance gains, despite both being derived from the same underlying input data, namely users' written reviews, indicating that each preserves distinct aspects of the preference signal. We observe similar improvements when ensembling text-based and interaction-based models, suggesting that data in free-form text and in fixed-form interactions complement each other.

Finally, the three-way ensemble combining text, categorical, and interaction representations achieves the best or statistically tied-for-best result (paired Wilcoxon test (43), $p > 0.05$) in every (dataset, metric) cell (shown for NDCG@10 in Figure 5 and for all four metrics in Table E.1). These results indicate that free-form text, categorical tags, and interaction data encode overlapping but non-redundant preference signals that can be effectively combined with simple ensembling.

5 Discussion and conclusion

This paper asks how much historical interaction data is needed to match the preference signal in a user's own words. Our comparison deliberately preserves the data advantage of interaction-based methods: collaborative filtering trains on abundant task-specific interactions and is heavily tuned, while text-based representations use an off-the-shelf zero-shot LLM without fine-tuning. Even so, collaborative filtering needs on the order of a million train interactions to match the off-the-shelf GPT-5 model given a few paragraphs about a user, condensed from a dozen or two of their reviews. This crossover point is likely to keep moving as LLMs improve: with the more recent Claude Opus 5.5, it already lies at a few million interactions.

Our results suggest that free-form text can stand in for large amounts of interaction data, well beyond the cold-start settings where it was traditionally found most useful. In economic prediction tasks, a pretrained LLM has been found to substitute for only hundreds of domain-specific observations (44), far fewer than the million interactions it matches here. This contrast suggests that preferences expressed in plain language are an unusually information-rich signal. We further show that an LLM can distill item descriptions into an open-ended set of interpretable categorical

features, and users' reviews into their stated preferences over them, which together predict choices as well as the text they come from. Free-form text, categorical features, and interactions are also complementary: combining them consistently yields the best results.

Risks and limitations. Several practical limitations warrant discussion. First, LLM-based scoring is expensive, although it is quickly getting cheaper: moving from GPT-5 to the more recent GPT-6-luna cuts API costs by 98% with little loss in performance. The categorical representation sidesteps most of this cost: it uses the LLM only once per user, to build a profile from their reviews, and scores each candidate item with a lightweight supervised model. Second, text brings risks of its own. LLMs can infer private attributes of a text's author (45), so stored text profiles raise privacy concerns, as interaction logs also do (46). An LLM scorer also reads third-party text, such as item descriptions, which exposes it to prompt injection aimed at manipulating its scores (47).

Lastly, our evaluation setup has limits of its own. It spans two consumer-review platforms (Amazon and Goodreads) and is restricted to users who authored reviews, who may differ from users who do not. Both corpora are also public and drawn from popular, heavily reviewed consumer domains: although our design prevents direct leakage and the evidence strongly argues against memorization driving the results, the LLM's general knowledge of these domains and their items likely contributes to its zero-shot performance, beyond the item information in its prompt. Both datasets predate the LLM's pretraining cut-off, so how the text representation fares on items released after that cut-off remains to be seen. The text should also capture the whole interaction between user and item, which is harder in some domains: while everyone reads the same book, a restaurant can offer a different experience on each visit.

At the same time, our setup favors interaction-based methods, so our crossover estimates are conservative. Like standard recommender-system benchmarks, it scores models on predicting held-out interactions, the very objective collaborative filtering is trained on. This objective is easy to evaluate, but it likely diverges from broader measures of user satisfaction and well-being (25). A better evaluation would measure these outcomes directly, for instance through controlled user experiments, and we expect it to push the crossover further out.

Choice models and mechanism design. Our comparison also relates to the economics of demand estimation. Collaborative filtering represents users and items through latent factors inferred from choices, whereas the categorical representation places them in a shared space of named characteristics. Both share the structure of Lancaster's classic model of consumer choice (21, 48), in which people value goods for their characteristics rather than the goods themselves. Characteristics are thus a natural unit of preference: they extend to goods no one has chosen yet, and they match how people describe what they want. In the estimation of choice models, a long-standing empirical challenge has been obtaining measures of the characteristics themselves, which have typically been hand-coded and low-dimensional, with each consumer's tastes over them inferred from repeated choices. In our framework, language supplies both, characteristics from item descriptions and tastes from users' own reviews, enabling a scalable alternative. Appendix B develops this connection and outlines a conditional logit over these characteristics, which would yield interpretable taste parameters and allow testing whether tastes stated in text add information beyond those inferred from choices.

Any elicitation interface is itself a mechanism (a set of rules mapping what people report to what they receive), and users respond to those rules when deciding what to report. Two classic design requirements follow. Users are more likely to share preference text if doing so improves their experience (a participation constraint), and will state their preferences honestly only if doing so works in their favor (incentive compatibility). Both are plausible for recommendation systems, where

stated preferences and the goal of recommending items that users value are broadly aligned. The user statements that we study also sidestep a core concern about stated preferences, hypothetical bias (22), because they are reviews written after consumption. These constraints play a bigger role in the allocation of scarce goods, where reported preferences determine who gets what and misreporting can pay. Our static evaluation leaves open how richer preference representations interact with such strategic behavior; emerging mechanisms that score free-text reports for informativeness offer candidate tools (49, 50). Whether a language interface elicits preferences without shaping them is a further open question.

Broader impact. These findings have broad implications for preference elicitation, recommendation, and beyond. The choice of representation shapes what systems optimize: engagement is easy to measure and to optimize directly from logged interactions, so engagement-maximizing feeds follow naturally from machinery built on fixed-form signals. Such feeds can exploit the gap between users' in-the-moment behavior and their considered preferences (15, 24). Preferences stated in a user's own words plausibly sit closer to considered judgment than behavioral traces do, and our results show that LLMs now make them competitive with abundant interaction data.

Stated preferences are also legible to the people they describe. Persistent "memory" features in LLM assistants provide a useful design precedent: the user's profile is a plain-language record that its owner can read, correct, and delete, whereas a learned model offers only opaque latent factors. Such a profile could also travel across platforms, and its owner could choose which actions inform it. Future preference-learning systems could provide similar control while retaining interactions as a complementary signal.

A text-first approach to representing user preferences can benefit both users (transparency and control) and platforms (predictive performance). It could be a first step away from the opaque, engagement-maximizing feeds that now shape much of what people see online, giving agency back to users. Our results support a simple design principle: ask users what they want, and let them answer in their own words.

Acknowledgments

The authors are grateful to Yatong Chen and Ricardo Dominguez-Olmedo for many discussions that shaped this work and for their feedback at every stage. We are grateful to Moritz Hardt, Ariel Procaccia, and Ana-Andreea Stoica for invaluable discussions that shaped the framing of this work, and to Jerry Green and Matthew Rabin for key economic insights.

We thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting André F. Cruz. Rediet Abebe thanks the Hector Foundation and the Andrew Carnegie Fellowship Program for supporting this work. We thank grants from the Simons Foundation and MacArthur Foundation for supporting Jon Kleinberg. This work was partially completed during research visit at the Harvard John A. Paulson School Of Engineering And Applied Sciences.

We used Anthropic's Claude Opus 4.5–5.5 to assist with methods implementation, plotting code, and manuscript reviewing. The authors reviewed all AI-assisted outputs and take full responsibility for the work.

References

[1] Ali Shirali, Jessie Finocchiaro, and Rediet Abebe. Participatory objective design via preference elicitation. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '24, pages 1637–1662, New York, NY, USA, 2024. Association for Computing Machinery. doi:10.1145/3630106.3658994.

[2] Yan Chen and Tayfun Sönmez. School choice: an experimental study. *Journal of Economic Theory*, 127 (1):202–231, 2006. doi:10.1016/j.jet.2004.10.006.

[3] Carlos A. Gomez-Uribe and Neil Hunt. The Netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems*, 6(4):1–19, December 2015. doi:10.1145/2843948.

[4] Kenneth Joseph Arrow. *Social Choice and Individual Values*. Wiley, New York, NY, USA, 1951.

[5] Yann Chevaleyre, Ulle Endriss, Jérôme Lang, and Nicolas Maudet. Preference handling in combinatorial domains: From AI to social choice. *AI Magazine*, 29(4):37–46, 2008. doi:10.1609/aimag.v29i4.2201.

[6] Samantha Robertson and Niloufar Salehi. What if I don't like any of the choices? The limits of preference elicitation for participatory algorithm design. *arXiv preprint arXiv:2007.06718*, 2020.

[7] Noam Nisan and Ilya Segal. The communication requirements of efficient allocations and supporting prices. *Journal of Economic Theory*, 129(1):192–224, 2006. doi:10.1016/j.jet.2004.10.007.

[8] Paul Dütting, Felix Fischer, and David C. Parkes. Simplicity-expressiveness tradeoffs in mechanism design. In *Proceedings of the 12th ACM Conference on Electronic Commerce*, EC '11, pages 341–350, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450302616. doi:10.1145/1993574.1993632.

[9] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272, 2008. doi:10.1109/ICDM.2008.22.

[10] Hal R. Varian. Revealed preference and its applications. *The Economic Journal*, 122(560):332–338, May 2012. ISSN 0013-0133. doi:10.1111/j.1468-0297.2012.02505.x.

[11] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. GroupLens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, CSCW '94, pages 175–186, New York, NY, USA, 1994. Association for Computing Machinery. doi:10.1145/192844.192905.

[12] G. Linden, B. Smith, and J. York. Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, 2003. doi:10.1109/MIC.2003.1167344.

[13] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009. doi:10.1109/MC.2009.263.

[14] Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 165–172, New York, NY, USA, 2013. Association for Computing Machinery. doi:10.1145/2507157.2507163.

[15] Jon Kleinberg, Jens Ludwig, Sendhil Mullainathan, and Manish Raghavan. The inversion problem: Why algorithms should infer mental state and not just predict behavior. *Perspectives on Psychological Science*, 19(5):827–838, 2024. doi:10.1177/17456916231212138. PMID: 38085919.

[16] Carey K. Morewedge, Sendhil Mullainathan, Haaya F. Naushan, Cass R. Sunstein, Jon Kleinberg, Manish Raghavan, and Jens O. Ludwig. Human bias in algorithm design. *Nature Human Behaviour*, 7 (11):1822–1824, November 2023. doi:10.1038/s41562-023-01724-4.

[17] Amanda Y Agan, Diag Davenport, Jens Ludwig, and Sendhil Mullainathan. Automating automaticity: How the context of human choice affects the extent of algorithmic bias. Working Paper 30981, National Bureau of Economic Research, February 2023. URL `http://www.nber.org/papers/w30981`.

[18] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. Recommendations as treatments: Debiasing learning and evaluation. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1670–1679, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/schnabel16.html.

[19] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, pages 253–260, New York, NY, USA, 2002. Association for Computing Machinery. doi:10.1145/564376.564421.

[20] Bailey Flanigan and Michelle Si. Internal pluralism and the limits of pairwise comparisons. *arXiv preprint arXiv:2607.02672*, 2026.

[21] Kelvin J. Lancaster. A new approach to consumer theory. *Journal of Political Economy*, 74(2):132–157, 1966. doi:10.1086/259131.

[22] Jordan J. Louviere, David A. Hensher, and Joffre D. Swait. *Stated Choice Methods: Analysis and Applications*. Cambridge University Press, 2000.

[23] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2): 179–188, 1936. doi:10.1111/j.1469-1809.1936.tb02137.x.

[24] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. The challenge of understanding what users want: Inconsistent preferences and engagement optimization. *Management Science*, 70(9): 6336–6355, 2024. doi:10.1287/mnsc.2022.03683.

[25] Smitha Milli, Micah Carroll, Yike Wang, Sashrika Pandey, Sebastian Zhao, and Anca D. Dragan. Engagement, user satisfaction, and the amplification of divisive content on social media. *PNAS Nexus*, 4 (3):pgaf062, 2025. doi:10.1093/pnasnexus/pgaf062.

[26] Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 73–105. Springer US, Boston, MA, 2011. doi:10.1007/978-0-387-85820-3_3.

[27] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, pages 83–92, New York, NY, USA, 2014. Association for Computing Machinery. doi:10.1145/2600428.2609579.

[28] Lei Zheng, Vahid Noroozi, and Philip S. Yu. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, pages 425–434, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450346757. doi:10.1145/3018661.3018665.

[29] María Hernández-Rubio, Iván Cantador, and Alejandro Bellogín. A comparative analysis of recommender systems based on item aspect opinions extracted from user reviews. *User Modeling and User-Adapted Interaction*, 29(2):381–441, 2019. doi:10.1007/s11257-018-9214-9.

[30] Noveen Sachdeva and Julian McAuley. How useful are reviews for recommendation? A critical review and potential improvements. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, pages 1845–1848, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3397271.3401281.

[31] Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. ChatGPT outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120, 2023. doi:10.1073/pnas.2305016120.

[32] Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 Embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

[33] OpenAI. How memory works. Memory in ChatGPT, https://help.openai.com/en/articles/8590148-memory-faq, 2026. Accessed: 2026-07-12.

[34] Ermis Soumalias, Yanchen Jiang, Kehang Zhu, Michael J. Curry, Sven Seuken, and David C. Parkes. LLM-powered preference elicitation in combinatorial assignment. *arXiv preprint arXiv:2502.10308*, 2025.

[35] Sara Fish, Paul Gölz, David Parkes, Ariel Procaccia, Gili Rusak, Itai Shapira, and Manuel Wuthrich. Generative social choice. *J. ACM*, 73(2), April 2026. doi:10.1145/3799709.

[36] Mengting Wan and Julian McAuley. Item recommendation on monotonic behavior chains. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18, pages 86–94, New York, NY, USA, 2018. Association for Computing Machinery. doi:10.1145/3240323.3240369.

[37] Yupeng Hou, Jiacheng Li, Xiangjun Fu, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. Bridging language and items for retrieval and recommendation: Benchmarking LLMs as semantic encoders. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens, editors, *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3251–3265, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi:10.18653/v1/2026.acl-long.147.

[38] James Bennett and Stan Lanning. The Netflix Prize. In *Proceedings of KDD Cup and Workshop 2007*, pages 3–6, New York, NY, USA, August 2007. ACM. URL http://www.cs.uic.edu/~liub/KDD-cup-2007/NetflixPrize-description.pdf.

[39] Steffen Rendle, Walid Krichene, Li Zhang, and John Anderson. Neural collaborative filtering vs. matrix factorization revisited. In *Proceedings of the 14th ACM Conference on Recommender Systems*, RecSys '20, pages 240–248, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3383313.3412488.

[40] Ben Frederickson. implicit: Fast Python collaborative filtering for implicit feedback datasets, 2016. Software available from https://github.com/benfred/implicit.

[41] Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. OpenAI GPT-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

[42] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, October 2002. ISSN 1046-8188. doi:10.1145/582415.582418.

[43] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945. doi:10.2307/3001968.

[44] Wayne Gao, Sukjin Han, and Annie Liang. How well do LLMs predict human behavior? A measure of their pretrained knowledge, 2026. URL https://arxiv.org/abs/2601.12343.

[45] Robin Staab, Mark Vero, Mislav Balunovic, and Martin Vechev. Beyond memorization: Violating privacy via inference with large language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun, editors, *International Conference on Learning Representations*, volume 2024, pages 33832–33878, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/9028b8a3ca98f58e373f0c1497a17448-Paper-Conference.pdf.

[46] Michal Kosinski, David Stillwell, and Thore Graepel. Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences*, 110(15):5802–5805, 2013. doi:10.1073/pnas.1218772110.

[47] Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, AISec '23, pages 79–90, New York, NY, USA, 2023. Association for Computing Machinery. doi:10.1145/3605764.3623985.

[48] Daniel McFadden. Conditional logit analysis of qualitative choice behavior. In Paul Zarembka, editor, *Frontiers in Econometrics*, pages 105–142. Academic Press, New York, NY, USA, 1974.

[49] Yuxuan Lu, Shengwei Xu, Yichi Zhang, Yuqing Kong, and Grant Schoenebeck. Eliciting informative text evaluations with large language models. In *Proceedings of the 25th ACM Conference on Economics and Computation*, 2024.

[50] Yifan Wu and Jason D. Hartline. ElicitationGPT: Text elicitation mechanisms via language models. *arXiv preprint arXiv:2406.09363*, 2024.

[51] Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah Smith, Mari Ostendorf, and Hannaneh Hajishirzi. Fine-grained human feedback gives better rewards for language model training. In *Advances in Neural Information Processing Systems*, volume 36, pages 59008–59033, 2023.

[52] Katherine M. Collins, Najoung Kim, Yonatan Bitton, Verena Rieser, Shayegan Omidshafiei, Yushi Hu, Sherol Chen, Senjuti Dutta, Minsuk Chang, Kimin Lee, Youwei Liang, Georgina Evans, Sahil Singla, Gang Li, Adrian Weller, Junfeng He, Deepak Ramachandran, and Krishnamurthy Dj Dvijotham. Beyond thumbs up/down: Untangling challenges of fine-grained feedback for text-to-image generation. *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 7:293–303, 2024. doi:10.1609/aies.v7i1.31637.

[53] Marquis de Condorcet. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. L'Imprimerie royale, Paris, 1785.

[54] Ariel D. Procaccia and Jeffrey S. Rosenschein. The distortion of cardinal preferences in voting. In Matthias Klusch, Michael Rovatsos, and Terry R. Payne, editors, *Cooperative Information Agents X*, pages 317–331, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[55] Belinda Z. Li, Alex Tamkin, Noah Goodman, and Jacob Andreas. Eliciting human preferences with language models. In *International Conference on Learning Representations*, 2025.

[56] David Huang, Edwin Lock, Francisco Marmolejo-Cossío, and David C. Parkes. Accelerated preference elicitation with LLM-based proxies. In Vittorio Bilò, Yang Cai, and Zhiyi Huang, editors, *Web and Internet Economics*, pages 631–648, Cham, 2026. Springer Nature Switzerland. ISBN 978-3-032-18660-7.

[57] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

[58] Chong Wang and David M. Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 448–456, New York, NY, USA, 2011. Association for Computing Machinery. doi:10.1145/2020408.2020480.

[59] Scott Sanner, Krisztian Balog, Filip Radlinski, Ben Wedin, and Lucas Dixon. Large language models are competitive near cold-start recommenders for language- and item-based preferences. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 890–896, 2023.

[60] Zhi Zheng, WenShuo Chao, Zhaopeng Qiu, Hengshu Zhu, and Hui Xiong. Harnessing large language models for text-rich sequential recommendation. In *Proceedings of the ACM Web Conference 2024*, WWW '24, pages 3207–3216, New York, NY, USA, 2024. Association for Computing Machinery. doi:10.1145/3589334.3645358.

[61] Damien Sileo, Wout Vossen, and Robbe Raymaekers. Zero-shot recommendation as language modeling. In *Advances in Information Retrieval: 44th European Conference on IR Research (ECIR 2022)*, pages 223–230. Springer, 2022.

[62] Koki Ryu and Hitomi Yanaka. Enhancing rating prediction with off-the-shelf LLMs using in-context user reviews. *arXiv preprint arXiv:2510.00449*, 2025.

[63] Paul F. Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf.

[64] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

[65] Angelica Chen, Jérémy Scheurer, Jon Ander Campos, Tomasz Korbak, Jun Shern Chan, Samuel R. Bowman, Kyunghyun Cho, and Ethan Perez. Learning from natural language feedback. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=xo3hI5MwvU.

[66] Ali Shirali, Arash Nasr-Esfahany, Abdullah Alomar, Parsa Mirtaheri, Rediet Abebe, and Ariel Procaccia. Direct alignment with heterogeneous preferences. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 69858–69898. Curran Associates, Inc., 2025. doi:10.52202/085713-2350.

[67] Luise Ge, Daniel Halpern, Evi Micha, Ariel D. Procaccia, Itai Shapira, Yevgeniy Vorobeychik, and Junlin Wu. Axioms for AI alignment from human feedback. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 80439–80465. Curran Associates, Inc., 2024. doi:10.52202/079017-2557.

[68] Vincent Conitzer, Rachel Freedman, Jobst Heitzig, Wesley H. Holliday, Bob M. Jacobs, Nathan Lambert, Milan Mossé, Eric Pacuit, Stuart Russell, Hailey Schoelkopf, Emanuel Tewolde, and William S. Zwicker. Position: Social choice should guide AI alignment in dealing with diverse human feedback. In *International Conference on Machine Learning*, volume 235, pages 9346–9360, 2024.

[69] Michael Henry Tessler, Michiel A. Bakker, Daniel Jarrett, Hannah Sheahan, Martin J. Chadwick, Raphael Koster, Georgina Evans, Lucy Campbell-Gillingham, Tantum Collins, David C. Parkes, Matthew M. Botvinick, and Christopher Summerfield. AI can help humans find common ground in democratic deliberation. *Science*, 386(6719):eadq2852, 2024.

[70] Taylor Sorensen, Jared Moore, Jillian Fisher, Mitchell L Gordon, Niloofar Mireshghallah, Christopher Michael Rytting, Andre Ye, Liwei Jiang, Ximing Lu, Nouha Dziri, Tim Althoff, and Yejin Choi. Position: A roadmap to pluralistic alignment. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 46280–46302. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/sorensen24a.html.

[71] Hannah Rose Kirk, Alexander Whitefield, Paul Röttger, Andrew Bean, Katerina Margatina, Juan Ciro, Rafael Mosquera, Max Bartolo, Adina Williams, He He, Bertie Vidgen, and Scott Hale. The PRISM alignment dataset: What participatory, representative and individualised human feedback reveals about the subjective and multicultural alignment of large language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 105236–105344. Curran Associates, Inc., 2024. doi:10.52202/079017-3342.

[72] HyunJi Nam, Yanming Wan, Mickel Liu, Peter Ahnn, Jianxun Lian, and Natasha Jaques. Learning to summarize user information for personalized reinforcement learning from human feedback. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust, editors, *International Conference on Learning Representations*, volume 2026, pages 122761–122784, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/c7af751b5e0a407c62ac023e3cb381f9-Paper-Conference.pdf.

[73] Rajiv Movva, Smitha Milli, Sewon Min, and Emma Pierson. What's in my human feedback? Learning interpretable descriptions of preference data. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang,

and A. Faust, editors, *International Conference on Learning Representations*, volume 2026, pages 59420–59455, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/613c72b51ebd8f44cd4126252348c624-Paper-Conference.pdf.

[74] Arduin Findeis, Timo Kaufmann, Eyke Hüllermeier, Samuel Albanie, and Robert Mullins. Inverse constitutional AI: Compressing preferences into principles. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, editors, *International Conference on Learning Representations*, volume 2025, pages 52687–52730, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/82eec786fdfbbfa53450c5feb7d1ac92-Paper-Conference.pdf.

[75] Henry Bell, Lara Neubauer da Costa Schertel, Bochu Ding, and Brandon Fain. *Learning Alignment Principles Grounded in Human Reasons and Values*, pages 6956–6997. Association for Computing Machinery, New York, NY, USA, 2026. ISBN 9798400725968. URL https://doi.org/10.1145/3805689.3806453.

[76] Jerome Ramos, Hossein A. Rahmani, Xi Wang, Xiao Fu, and Aldo Lipani. Transparent and scrutable recommendations using natural language user profiles. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13971–13984, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi:10.18653/v1/2024.acl-long.753.

[77] Joyce Zhou, Yijia Dai, and Thorsten Joachims. Language-based user profiles for recommendation. *arXiv preprint arXiv:2402.15623*, 2024.

[78] Manuel Trajtenberg. The welfare analysis of product innovations, with an application to computed tomography scanners. *Journal of Political Economy*, 97(2):444–479, 1989.

[79] Amil Petrin. Quantifying the benefits of new products: The case of the minivan. *Journal of Political Economy*, 110(4):705–729, 2002.

[80] Sherwin Rosen. Hedonic prices and implicit markets: Product differentiation in pure competition. *Journal of Political Economy*, 82(1):34–55, 1974. doi:10.1086/260169.

[81] Steven Berry, James Levinsohn, and Ariel Pakes. Automobile prices in market equilibrium. *Econometrica*, 63(4):841–890, 1995. doi:10.2307/2171802.

[82] Patrick Bajari and C. Lanier Benkard. Demand estimation with heterogeneous consumers and unobserved product characteristics: A hedonic approach. *Journal of Political Economy*, 113(6): 1239–1276, 2005. doi:10.1086/498586.

[83] Susan Athey, David Blei, Robert Donnelly, Francisco Ruiz, and Tobias Schmidt. Estimating heterogeneous consumer preferences for restaurants and travel time using mobile location data. *AEA Papers and Proceedings*, 108:64–67, 2018. doi:10.1257/pandp.20181031.

[84] Francisco J. R. Ruiz, Susan Athey, and David M. Blei. SHOPPER: A probabilistic model of consumer choice with substitutes and complements. *The Annals of Applied Statistics*, 14(1):1–27, 2020. doi:10.1214/19-AOAS1265.

[85] Matthew Gentzkow, Bryan Kelly, and Matt Taddy. Text as data. *Journal of Economic Literature*, 57(3): 535–574, 2019.

[86] Giovanni Compiani, Ilya Morozov, and Stephan Seiler. Demand estimation with text and image data. *The RAND Journal of Economics*, 2026. doi:10.1111/1756-2171.70052.

[87] Patrick Bajari, Zhihao Cen, Victor Chernozhukov, Manoj Manukonda, Suhas Vijaykumar, Jin Wang, Ramon Huerta, Junbo Li, Ling Leng, George Monokroussos, and Shan Wang. Hedonic prices and quality adjusted price indices powered by AI. *Journal of Econometrics*, 251:106052, 2025. doi:10.1016/j.jeconom.2025.106052.

[88] Nikolay Archak, Anindya Ghose, and Panagiotis G. Ipeirotis. Deriving the pricing power of product features by mining consumer reviews. *Management Science*, 57(8):1485–1509, 2011. doi:10.1287/mnsc.1110.1370.

[89] Jens Ludwig, Sendhil Mullainathan, and Ashesh Rambachan. Large language models: An applied

econometric framework. *Annual Review of Economics*, 18(1):283–316, 2026. doi:10.1146/annurev-economics-120925-105620.

[90] Daniel McFadden. Modelling the choice of residential location. In Anders Karlqvist, Lars Lundqvist, Folke Snickars, and Jörgen W. Weibull, editors, *Spatial Interaction Theory and Planning Models*, pages 75–96. North-Holland, Amsterdam, The Netherlands, 1978.

[91] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications. Wiley, New York, 1991. URL https://books.google.pt/books?id=CX9QAAAAMAAJ.

Appendix

A Related work

We draw on three lines of work. The first studies how preferences are elicited and represented, the limitations imposed by constrained reporting languages, and recent efforts in mechanism design and social choice to admit natural-language reports via LLMs. The second examines the use of text as a signal in recommender systems, from early sparse representations to recent LLM-based approaches. The third studies the alignment of LLMs with human preferences, where the format of feedback and the heterogeneity of annotators have become central concerns. We review each in turn and highlight the empirical gap our work addresses.

Preference elicitation and representation. Preference elicitation has long balanced expressiveness against usability and scalability. In mechanism design and social choice, the chosen preference language (for example, rankings, cardinal utilities, or restricted reports) is itself a design decision that determines which welfare-relevant trade-offs can be communicated and optimized. For surveys of representation in combinatorial domains see Chevaleyre et al. (5), and for formal treatments of simplicity versus expressiveness see Dütting et al. (8).

A central consequence is that many real preferences cannot be faithfully captured by fixed-form primitives such as likes and dislikes. Binary feedback cannot express conditional or multi-attribute constraints that are natural in combinatorial models (5); examples include "I like political thrillers only when they are not graphic" or "I prefer slower books unless they are part of a long series." More broadly, representational choices and elicitation interfaces can introduce systematic biases by constraining what users are able or invited to report (6). These limits persist for modern foundation models: reward models are typically trained on coarse thumbs-up/down or ranking judgments, and although finer-grained feedback can improve reward learning (51), eliciting and modeling it reliably remains challenging (52). Flanigan and Si (20) formalize this limitation: modeling individuals as weighing multiple internal priorities, they show that priorities which evaluate outcomes globally rather than choice by choice (for example, egalitarian or equal-treatment concerns) are provably unidentifiable from local pairwise comparisons, and that requiring a decisive answer conflates genuine indifference with internal conflict, further distorting what is learned. Their proposed remedy is to elicit the underlying priorities directly, through text and structured queries; our results quantify how much preference signal such a text channel carries in practice.

At the aggregation level, classic impossibility results show further limits: even when individual preferences are transitive, no aggregation rule can satisfy a set of seemingly minimal desiderata simultaneously (4). The Condorcet paradox illustrates this tension (53). Consider three alternatives A, B, C and three individuals with transitive preferences:

$$\succ_1\colon\ A \succ B \succ C, \quad \succ_2\colon\ B \succ C \succ A, \quad \succ_3\colon\ C \succ A \succ B.$$

Under pairwise majority aggregation, $A \succ B$ since $A \succ_1 B$ and $A \succ_3 B$, with analogous majorities yielding $B \succ C$ and $C \succ A$. Hence the social relation cycles as $A \succ B \succ C \succ A$, despite each $\succ_i$ being transitive. This example highlights that ordered lists and other restricted reports can fail to preserve structure that matters for aggregation and normative objectives (54).

A related distinction is between revealed and stated preferences. Interaction logs are typically treated as revealed preferences, but they are entangled with exposure and platform dynamics,

complicating inference about true user goals (17, 24). Stated-preference methods instead elicit counterfactual choices or structured reports; see Louviere et al. (22) and the review by Varian (10). Language-based profiles can be seen as a stated-preference channel that additionally carries explanations and conditional constraints that are invisible in interaction traces (6).

These representational and aggregation limitations motivate richer elicitation channels, including free-form text, which can convey nuances that fixed-form signals cannot. Li et al. (55) show that LLMs can drive such elicitation themselves: free-form questioning by a language model yields preference specifications more informative than user-written prompts or labeled examples.

Natural-language elicitation in mechanism design and social choice. A recent line of work acts on this motivation directly, admitting natural language as the report format of the mechanism itself. In combinatorial assignment, Soumalias et al. (34) have each student describe their course preferences once in free-form text, with an LLM proxy answering the mechanism's iterative comparison queries on the student's behalf; Huang et al. (56) apply similar proxies to accelerate bidder elicitation in combinatorial auctions. In social choice, Fish et al. (35) use LLMs to aggregate free-text opinions into proportionally representative slates of statements. A separate thread designs incentives, extending peer prediction (49) and proper scoring rules (50) to textual reports scored by LLMs. These designs presuppose that natural-language reports carry enough preference information to drive allocation and aggregation, a premise so far tested in small-scale pilots or with simulated participants. Our results quantify that preference signal at scale on organic, user-authored text, measured against the abundant interaction data platforms already collect.

Textual preference signals in recommender systems. Text has been used in recommender systems for decades, but early NLP pipelines typically reduced language to sparse tabular signals. Classic weighting schemes such as TF–IDF (57) and topic-model hybrids (e.g., 58) enabled scalable learning from reviews, and models by McAuley and Leskovec (14) and Zhang et al. (27) used text to surface latent factors or explainable dimensions. Nevertheless, these approaches relied on brittle feature extraction and limited semantic generalization, so text was often relegated to side information in cold-start scenarios, not yet viable as a first-class preference representation (26).

Recent advances in large language models change this picture by enabling direct interpretation of unstructured user-authored text, without task-specific feature engineering. In controlled evaluations, Sanner et al. (59) find LLMs competitive near the cold start regime, and Zheng et al. (60) demonstrate that LLMs can effectively leverage text-rich signals in sequential recommendation. Close to our comparison, existing work finds that GPT-2 prompted with movie titles beats matrix factorization only when the latter is trained on fewer than 100 users (61). More recently, off-the-shelf LLMs reading a user's reviews performed comparably on rating prediction to matrix factorization trained on a few thousand ratings (62). Still, most controlled studies abstract away key operational realities: they use limited interaction training data, and rely on experimentally elicited preference text that is denser and cleaner than organic free-form inputs.

Alignment and preference heterogeneity. LLM alignment is itself a preference learning problem built on the fixed-form signals discussed above: standard methods train on pairwise judgments pooled across annotators (63, 64). This design faces two distinct problems. First, the feedback format limits what each person can express: pairwise comparisons convey limited information about preferences (20), and related work suggests that learning from both language and comparison feedback outperforms learning from either alone (65). Second, pooling treats the population's preferences as homogeneous: under heterogeneity, direct alignment can optimize

a Borda-style aggregate rather than the population-average reward, and recovering the latter requires annotator-linked information (66); axiomatic analyses likewise show that standard reward learning violates basic social-choice desiderata (67), and Conitzer et al. (68) argue that social choice theory should guide how such diverse feedback is aggregated. One response uses language itself to aggregate: LLMs can synthesize representative positions from free-text opinions (35) and help groups with conflicting views draft consensus statements (69). A second response personalizes alignment, representing multiple values or conditioning model behavior on individual users (70); datasets such as PRISM preserve annotator profiles alongside their feedback for this purpose (71). Closest to our work, Nam et al. (72) summarize each user's preferences and conversation history into text that conditions a personalized reward model. Our results provide complementary evidence from organic recommendation data: user-authored language is an informative per-user preference representation. Our ensembles of text and interaction signals achieve the strongest overall performance, mirroring the complementarity of language and comparison feedback in alignment (65).

Interpretable representations of text preference data. A natural follow-up question is whether rich textual feedback can be compressed into structured representations without losing its signal. Within the alignment literature, a growing body of work pursues this question: instead of treating free-form feedback only as training data, these approaches extract structured, interpretable representations that can be inspected and audited. One recurring theme shows that preference datasets often admit low-dimensional, human-interpretable representations, where a small number of latent or structured features capture much of the signal used by dense reward models (73). A second line frames alignment as a compression problem. It distills large collections of pairwise judgments into concise natural-language principles or constitutions that reconstruct preferences and surface systematic biases (74); grounding these principles in the reasons people give for their preferences yields constitutions that study participants preferred (75). Complementary efforts decompose preferences into sociolinguistic or multi-attribute dimensions. Related work in recommendation builds editable natural-language user profiles to improve transparency and scrutability (76, 77). Together, these directions establish that unconstrained textual feedback can be transformed into compact, interpretable structures. This motivates our study of whether similar LLM-enabled categorical feature extraction preserves preference signal in large-scale, organically generated preference data.

Gap and contribution. Existing work establishes that text can encode meaningful preference information beyond fixed-form structured data. Evidence from both recommender systems (59, 76) and LLM alignment (73, 74) shows that explicitly elicited preference text can support rich, and often interpretable, representations. However, these results are largely obtained in controlled settings with clean, purpose-collected text, and do not address a key asymmetry in deployed systems: abundant historical interaction data versus organically generated preference text. We provide the first head-to-head comparison with modern LLMs of free-form preference text and structured preference representations under realistic conditions, using benchmark datasets drawn from real-world platforms. We show that a zero-shot text predictor remains competitive until collaborative filtering is trained on roughly one million interactions, and that much of text's signal can be distilled into interpretable, LLM-extracted categorical features.

B The choice-model connection

B.1 The Lancaster–McFadden framework for preferences over characteristics

A classical model of consumer demand, proposed by Lancaster (21), holds that people value goods for their characteristics, not the goods themselves. A central difficulty has been measuring those characteristics; our method provides a new way of supplying them from language. For example, a book can be desirable not because it is a particular book but because it has certain features or qualities (e.g., being fast-paced or having rich world-building). In our pipeline these qualities are recovered as tags read from what users write. Consumers thus prefer some qualities and characteristics over others, and value goods according to their characteristics.

The framework has two components. The first is what Lancaster refers to as a *consumption technology*, which maps goods to the characteristics they possess. Readers of books may disagree about whether they *want* a fast-paced book, but the framework treats whether a book *is* fast-paced as a property of the book. A key assumption here is that everyone perceives the characteristics the same way, i.e., all consumers face the same technology. In reality, some characteristics are close to objective, such as whether a book belongs to a specific series, while others are contested. This assumption is standard in the literature. In our context, it means that goods carry a shared characteristics set, inferred by an imperfect model that induces some error. §B.4 discusses the implications. The second component is how each consumer attaches weights to those characteristics, where taste enters and consumers differ.

Because preferences attach to characteristics of the goods, and not the goods themselves, the framework enables predictions even for items that have never been chosen or observed. A new good may simply be a bundle of familiar characteristics. An unseen good therefore gets predictions close to those of familiar goods with similar characteristics. In this way, the original framework of Lancaster anticipated what recommender systems now call cold-start.

McFadden (48) supplied a tractable random-utility framework for estimating discrete choice over observed characteristics. Specifically, adding an idiosyncratic shock to characteristics-based utility, with the shock following a type-I extreme value distribution, yields the conditional logit model. When this model is applied to choice data it identifies the weights on characteristics.[1] This framework has been used to empirically value new goods introduced into a market from the attributes they combine, such as the CT scanner (78) and the minivan (79).

How to infer and populate the characteristics space has been a long-standing empirical bottleneck. In applied work it has typically been hand-coded, small, and chosen by the analyst (79, 80), and the gap between the attributes an econometrician records and those a consumer perceives is why demand models must leave room for characteristics they cannot see (81). Recovering those hidden characteristics has itself become a line of research (82). Our approach offers a scalable way to infer characteristics space of goods from text. We recover them from language, learning a space of roughly ten thousand characteristics that is large, interpretable, and predictive. In Lancaster's terms, our item-tagging step estimates the consumption technology that maps from goods to characteristics that his framework takes as given and that the analyst has otherwise supplied by hand. Clustering this space into fewer characteristics (§E.7) further eases interpretability, at a modest cost to the categorical model (around 0.03 validation NDCG@10 with 1,000 clusters and 0.06 with 100).

[1] The discrete-choice version also avoids the divisibility and corner-solution problems of Lancaster's linear formulation.

B.2 Collaborative filtering as a latent-characteristics model

The models we compare all assign utility in an interplay between a user-level vector and item-side vector, but differ vastly in how they get computed and their meaning. The categorical MLP first computes the element-wise product $x_{ut}x_{it}$, the overlap between the user's vocabulary and the item's, tag by tag. These overlaps along with the absolute difference, $|x_{ut} - x_{it}|$, are inputs to a neural network that computes the relevance score. Collaborative filtering works quite differently, but shares the same interaction between user-side and item-side representations. Its score is the inner product $\langle \mathbf{U}_u, \mathbf{V}_i \rangle = \sum_d U_{ud} V_{id}$, the same overlap-and-weight operation over dimensions that are neither named nor chosen. The conditional logit is arguably the simplest model using the same interplay. It replaces the neural network structure of the MLP with simple set of weights of the characteristics, one taste coefficient β_t per characteristic. This simplicity also makes the conditional logit more interpretable. We outline this model in §B.4.

In economics, an inner-product score becomes a multinomial logit once model weights are estimated by maximizing a softmax likelihood over a choice set (48). For matrix factorization, the alternative attributes are the latent item factors, $\mathbf{V}_i$, and the coefficient vector, $\mathbf{U}_u$, holds the user's preferences over them. Latent-factor choice models of this kind have been used for restaurant choice and grocery purchases (83, 84).

We do not train collaborative filtering as a conditional logit and want to highlight some differences. We use the standard implicit-feedback estimator (9). This approach minimizes a confidence-weighted squared error over the full user-item matrix instead of a likelihood. Observed user-item pairs get assigned the weight $1 + \alpha$ and every unobserved pair is assigned unit weight. The negatives are not drawn during training, because every item the user has not interacted with enters the scoring as a weak negative. Here, the confidence weight α thus plays a similar role to the positive-to-negative weighting stemming from sampling negative alternatives. Thus, the model optimizes over the full user-item matrix rather than sampled alternatives, using an unnormalized loss rather than a choice probability. Sampled negatives enter our design only in the evaluation candidate sets and in fitting the categorical scorer. Finally, the scoring uses squared error on a binary target, so the estimator is closer to a linear probability model over latent characteristics than to a logit.

On the user side, collaborative filtering also infers the users' taste parameters from behavior. This is done by running a regularized regression of profile interactions on item factors that are held fixed. For the categorical representation there is an analogous user-side representation, $\mathbf{x}_u$, that comes from an LLM reading what the user wrote. The users profile interactions enter the categorical model only as training pairs for the shared MLP weights. For both models, the inputs serve as user-side taste vectors, one inferred from choices and one extracted from the text.

From this perspective, the collaborative filtering and categorical models are two distinct ways of constructing representations of users and items. One uses latent characteristics inferred from behavior while the other uses named characteristics measured from language. What we contribute is not a new framework for choice modeling but an approach for inferring interpretable characteristics rather than latent ones. These characteristics can be applied in existing discrete choice frameworks.

B.3 Related approaches for text and embeddings in demand estimation

The use of text as data is already established in economics (85). The applications to demand estimation have primarily used text and unstructured data to describe the goods.

Compiani et al. (86) extract embeddings from product descriptions and images. These embeddings

are used as product characteristics in a mixed logit, recovering substitution patterns that hand-coded attributes can miss. Bajari et al. (87) extract attributes from text and images with neural networks to estimate hedonic price indices on Amazon apparel. Earlier, Archak et al. (88) mined product features and consumer opinions from reviews and estimated the pricing power of individual features from sales-rank data. The three studies differ in method and in target, but they all use text as a source of item characteristics; the consumer enters only through their choices.

Our setting has text on both sides. Item characteristics come from descriptions and community reviews, as in the work above; user characteristics come from that user's own writing, expressed in the same vocabulary. To our knowledge, the closest work in demand estimation has focused on item-side characteristics whereas our approach infers both user and item characteristics from text.

Our approach also differs in how text is represented. Unlike the embeddings used in the studies above, our named characteristics make coefficients interpretable, though only to the extent that the specific tags are meaningful. The extraction process is therefore crucial, since a coefficient means something only if the characteristic is measured correctly (89), an issue we return to in §B.4.

B.4 A Lancaster-style conditional logit

Because the categorical representation places users and items in the same space of named characteristics, a conditional logit can be fitted directly on it, with one coefficient per characteristic. Write the utility user u derives from item i as

$$u_{ui} = \gamma^\top \mathbf{x}_i + \beta^\top (\mathbf{x}_u \odot \mathbf{x}_i) + \varepsilon_{ui}, \tag{1}$$

where γ_t is the average marginal utility from characteristic t and β_t the additional utility when the user's own text also carries tag t. Personalization is then a pure interaction effect. Econometrically this is a conditional logit with observed heterogeneity, where the user's tags play the role that sociodemographics play in McFadden's original applications, except that they are extracted from the user's own prose instead of using, e.g., recorded age or income.

To interpret the categorical representation within a random utility framework, we treat each observed interaction as arising from a latent choice occasion. The feasible choice set is approximated by the full available collection of items, including items that were not yet available at the time of the interaction. This constructed choice set may not capture consumers' actual consideration sets, and should be interpreted rather as a probabilistic approximation.

Our general utility function in Equation 1 nests a simpler baseline model with no role for the user's stated preferences over tags. If we set $\beta = 0$ the model becomes an item-side only logit that captures average tastes over characteristics. Running a statistical test of $\beta = 0$ therefore asks whether stated preferences carry information beyond average tastes over characteristics. That restricted model is also what would remain for a user who has written nothing (a cold start).

The characteristics are measured variables from LLMs. In prediction the goal is ranking items accurately, and interpretation is secondary. Conversely, for the conditional logit researchers are likely to be primarily interested in interpreting the coefficients and doing inference on them. Thus, measurement error in the tags a user likes, $\mathbf{x}_u$, and item tags, $\mathbf{x}_i$, can propagate into the estimated coefficients $\hat{\gamma}$ and $\hat{\beta}$. Interpreting these coefficients therefore requires several checks. First, a validation sample of human-checked user–tag pairs is needed (89). Second, the estimates should be shown to be robust to rewording the extraction prompt and to changing the extraction model.

We do not estimate the model in this paper. Estimating the model involves a number of modeling choices that we leave for future work.[2] This framework for modeling choice and integrating characteristics from text is intended as an interpretable probabilistic representation of the recommendation task and not as a fully structural demand model.

C Datasets

This appendix provides additional details on the six datasets used throughout the paper: three genre subsets of the `Goodreads.com` book platform (36) and three `Amazon.com` (37) product-review subsets. Its purpose is to document dataset composition, preprocessing, and evaluation splits in sufficient detail to support reproducibility, while keeping the main text focused on modeling and empirical comparisons. We also include exploratory analyses that help contextualize the observed performance of different baselines.

C.1 Dataset structure and interaction types

The original Goodreads dataset contains multiple forms of user–item interactions, including reading events, ratings, and free-form reviews. These interaction types differ in both coverage and semantic meaning, and raw interaction counts therefore differ from the number of books that were explicitly read or rated. Throughout the paper, we distinguish between raw interaction records and the cleaned interactions used for modeling. Raw records include a read indicator (and, when present, start/end dates), an explicit star rating, and an optional free-form review text. A record may contain any subset of these fields, so review counts are lower than read/rating counts.

C.2 Data preprocessing and filtering

We apply a set of conservative filtering steps to improve data quality and ensure a consistent interpretation of user–item interactions. Specifically, we retain only records with a non-null, strictly positive rating and a positive read indicator. To remove implausible or incomplete reading events, we require both a recorded start date and completion date, with at least two calendar days elapsed between them. For review text, we discard short reviews and retain only reviews exceeding a minimum character-length threshold of 500 characters.

After filtering, each remaining interaction corresponds to a book that was both read and explicitly rated by the user. Unless otherwise stated, all reported interaction statistics and experimental results are computed on this filtered dataset.

C.3 Goodreads genre-specific subsets

Experiments are conducted on three genre-specific subsets: *"Fantasy & Paranormal"*, *"Mystery, Thriller & Crime"*, and *"History & Biography"*. Each subset is constructed by restricting books to the corresponding genre labels and retaining all associated filtered interactions and reviews.

[2] A non-exhaustive list of problems to be dealt with includes: sampling of non-chosen alternatives and restricting these to items available at the time of the choice; applying the correction that popularity-weighted sampling requires (90); estimating on distinct clusters of the tag space where each coefficient has a name; fitting on training users so that no parameter is fit on a user the model is later scored on.

Tables C.1 and C.3 summarize the resulting dataset sizes after filtering, together with the user-level train/validation/test split; both also report the three Amazon subsets introduced in §C.4.

C.4 Amazon product-review subsets

We complement Goodreads with three Amazon product-review subsets: *Digital Music*, *Movies & TV*, and *Toys & Games*. These provide an analogous substrate of user-written reviews and user–item interactions in a distinct domain; here, an interaction corresponds to a product that was both purchased and rated by the user. We apply the same cleaning, filtering, and split protocol used for Goodreads. Tables C.1 and C.3 report split statistics for all six subsets.

C.5 Statistics by representation

The three representations are all derived from the same filtered interactions, yet they differ sharply in how much signal they expose per user and in which data their models consume. Of the three, only two involve trained models: collaborative filtering trained on raw interactions and a supervised MLP on categorical tag features. The text model is used zero-shot. Table C.1 therefore reports collaborative filtering's training data and Table C.2 the supervised MLP's training data. Table C.3 reports the profile and scoring data of the held-out evaluation users. The categorical feature vocabulary is detailed separately in Table D.1 (§D). Each table characterizes the data exactly as consumed by the corresponding models. Collaborative filtering is trained on size-capped subsets of the full train pool (1K to 1M interactions; Table C.1), matching the training-set sizes studied in the main text, with the selected configuration using 1M interactions. The supervised MLP instead trains on the evaluation population itself as the evaluation users have the necessary text data to generate categorical tags. The *profile* interactions of all 1,000 validation users and 1,000 test users are used to fit the MLP (39K–85K interactions; Table C.2). Naturally, test users' withheld interactions are not used in any way. The text representation involves no training: each evaluated user's profile is built from their reviews of at least 500 characters and consumed zero-shot by the LLM (Table C.3), where profile reviews coincide with profile interactions by construction. These profile reviews run longer on Goodreads than on Amazon (median 190–250 versus 142–185 words on test users). At evaluation time, each user contributes a single ranking task over a fixed candidate set of 50 items: exactly 10 held-out positive interactions together with 40 popularity-weighted negatives (a 20% positive rate). For each user, each of these 50 items is scored by all three models, taking the different representations as input. A few test users are not scored: the API's content filter blocked profile generation for 1–15 test users per dataset, flagging content in their reviews as violating the provider's usage policies. Without a profile, these users have no text or categorical representation, so the text and categorical results cover the remaining 985–999 test users per dataset.

C.6 Predictive power of items' popularity

Figure C.1 shows that users with longer interaction histories tend to consume, on average, less popular items. This observation helps explain why simple popularity-based baselines perform competitively for users with very short profiles, but deteriorate rapidly as profile length increases. As users become more data-rich, their behavior shifts toward the long tail of the catalog, where popularity alone is a weaker predictor of preferences.

Table C.1: **Collaborative-filtering training-set sizes.** Distinct users and items in the collaborative-filtering train user–item matrix at each training-set size used in the paper (the text model is zero-shot and has no training data; the MLP's is in Table C.2). Training sets accumulate whole users in random order until the target interaction count is met, overshooting it by at most 2.5% (1K), 1.2% (10K), 0.4% (100K), and 0.01% (1M) across datasets. The full eligible pool (gray) is shown for reference only, as no model in the paper is trained on it directly.

	Train-set size (interactions)										
	1K		**10K**		**100K**		**1M**		**Full pool**		
Dataset	Users	Items	Users	Items	Users	Items	Users	Items	Users	Items	Interactions
GR Fantasy	110	867	730	5.3K	7.4K	27.0K	74.2K	94.2K	337.7K	155.7K	4.55M
GR Thriller	117	847	1.0K	6.6K	11.7K	30.5K	116.8K	97.2K	275.6K	127.9K	2.35M
GR History	129	876	1.3K	6.3K	13.0K	33.1K	129.6K	117.5K	321.6K	163.2K	2.47M
AZ Music	348	985	4.1K	9.1K	43.3K	63.3K	431.7K	302.3K	1.35M	571.2K	3.14M
AZ Movies&TV	428	978	4.3K	8.5K	40.9K	49.8K	417.7K	169.4K	2.13M	317.5K	5.08M
AZ Toys&Games	534	1.0K	5.3K	8.8K	53.6K	58.4K	542.3K	242.6K	6.71M	686.1K	12.33M

Table C.2: **Per-user and per-item degree, and supervised-MLP training data.** For collaborative filtering, interaction degree (mean / median) at the 1M size used in the main comparisons; for the categorical MLP, the training set spans the 2,000 evaluation users' profile interactions (1,000 validation + 1,000 test). "Int./user" is the per-user count of these positive training interactions (mean / median over the combined population) and "Total int." their sum. The degree distributions are heavy-tailed, so the median is well below the mean, and on the Amazon subsets the median train user contributes a single train interaction at every size.

	Interactions (CF), at 1M		**Categorical (MLP)**	
Dataset	Int./user (mean / med.)	Int./item (mean / med.)	Int./user (mean / med.)	Total int.
GR Fantasy	13.5 / 4	10.6 / 2	42.5 / 27	85.0K
GR Thriller	8.6 / 3	10.3 / 2	36.7 / 22	73.5K
GR History	7.7 / 3	8.5 / 2	33.1 / 22	66.3K
AZ Music	2.3 / 1	3.3 / 1	31.8 / 20	63.7K
AZ Movies&TV	2.4 / 1	5.9 / 2	30.3 / 19	60.7K
AZ Toys&Games	1.8 / 1	4.1 / 1	19.6 / 17	39.3K

Table C.3: **Evaluation-user profiles.** Per-user counts and lengths of the profile reviews that build each held-out evaluation user's representation, for test and validation users separately; each profile cell reports mean / median. Every interaction carrying a review of at least 500 characters forms the *profile* (so "Profile int. ≡ reviews"); the remaining interactions form the held-out pool, which must hold at least 10 interactions. The text-based LLM consumes the entire profile zero-shot (train-split reviews appear only as item-side community-review context in scoring prompts; §F). "Held-out eval" is the candidate set scored for every evaluation user: 10 held-out positives (pos^+, a seeded random sample of the held-out pool) and 40 popularity-weighted negatives (neg^-).

	Test users			**Validation users**			**Held-out eval**
Dataset	Profile int./user (≡ reviews)	Review words	Review chars	Profile int./user (≡ reviews)	Review words	Review chars	(pos^+/neg^-)
GR Fantasy	23.6 / 12	316 / 250	1754 / 1370	21.7 / 12	318 / 240	1772 / 1329	$10^+/40^-$
GR Thriller	19.5 / 10	249 / 190	1399 / 1055	19.4 / 11	260 / 203	1461 / 1131	$10^+/40^-$
GR History	17.0 / 10	271 / 196	1543 / 1110	20.2 / 11	283 / 209	1617 / 1191	$10^+/40^-$
AZ Music	19.9 / 10	261 / 185	1518 / 1058	21.8 / 11	244 / 178	1411 / 1011	$10^+/40^-$
AZ Movies&TV	17.2 / 10	247 / 171	1422 / 966	19.4 / 9	244 / 169	1403 / 943	$10^+/40^-$
AZ Toys&Games	10.3 / 7	186 / 142	1010 / 759	10.5 / 7	185 / 144	1001 / 770	$10^+/40^-$

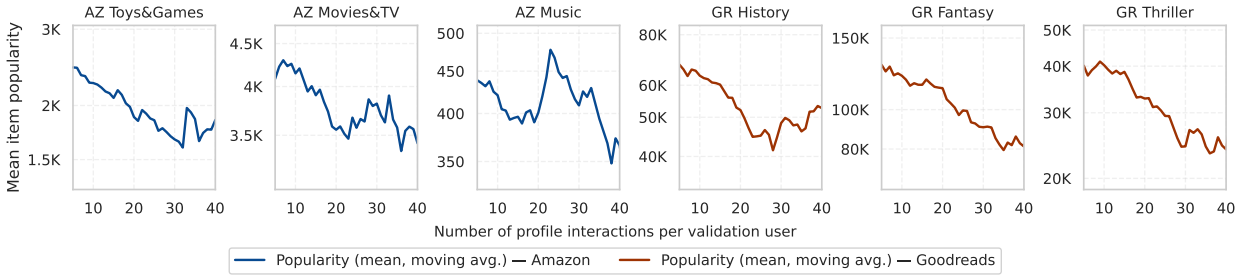


Figure C.1: **[All six datasets, Validation]** Mean popularity of the items users interacted positively with, as a function of each user's profile length (number of profile interactions), shown up to 40 profile interactions (covering 87–97% of validation users, depending on the dataset). Computed using a sliding window of ±10 profile interactions. Item popularity is measured as the total number of ratings an item has received on the platform; note the per-panel log scale.

D Categorical feature extraction details

This section details how categorical preference features are derived from user-authored text. For each user, an LLM generates free-form preference tags from their profile reviews; all generated tags are embedded with `Qwen3-Embedding-8B` (32), optionally consolidated into clusters by cosine similarity. Model selection on validation NDCG@10 retains the unclustered full tag vocabulary on all six datasets (clustering never improves on it; §E.7). Items are embedded into the same space and assigned their 5 nearest tags by cosine similarity. A small supervised MLP, fitted on the profile interactions of all 1,000 validation and 1,000 test users (39K–85K positive interactions per dataset; Table C.2), maps the element-wise product and absolute difference of the user's and item's categorical feature vectors, $\mathbf{x}_u$ and $\mathbf{x}_i$, i.e., $[\,\mathbf{x}_u \odot \mathbf{x}_i\,;\,|\mathbf{x}_u - \mathbf{x}_i|\,]$, to a relevance score. To form the binary training set, these positive pairs are augmented with sampled negatives: the selected final configurations use popularity-weighted sampling at 5, 10, 20, or 50 negatives per positive, yielding roughly 300K–2.0M negative pairs in the final MLP fits. Table D.1 reports the induced tag vocabulary. Across datasets the LLM generates between 8K and 14K distinct user tags in the merged validation+test population scored by the final model; because validation selection retains the unclustered vocabulary, these tags define the final feature dimensions used to represent both users' preferences and items' characteristics. Each item is assigned its 5 nearest tags. Because no clustering is applied, each tag is its own feature and users span the 15–17 tags they were assigned, so the feature dimension equals the full 8K–14K distinct user tags.

Table D.1: **Categorical feature vocabulary.** LLM-generated tag vocabulary, over the merged validation+test population scored by the selected model; the model's training-set size is reported in Table C.2. Validation NDCG@10 selects the unclustered full vocabulary on all six datasets, so the feature dimension equals the number of distinct user tags; users span ≈ 15–17 tags.

Dataset	Tag clusters	Distinct user tags	Tags /user	Distinct item tags
GR Fantasy	Full	14.1K	17.3	3.83K
GR Thriller	Full	12.8K	16.7	2.80K
GR History	Full	14.3K	16.5	4.42K
AZ Music	Full	9.74K	17.3	5.75K
AZ Movies&TV	Full	7.93K	15.4	4.50K
AZ Toys&Games	Full	8.67K	16.4	4.71K

E Additional results and ablations

This section reports additional results and ablations used to set key modeling choices. Additional results in §E.1, §E.2, §E.3, §E.4, §E.5, and §E.6 correspond to Test data. Hyperparameter-search results in §E.7 correspond to validation data: we select a single configuration per method based on these results before reporting final test performance in the main paper.

E.1 Full model comparison

This subsection reports the full model comparison. Table E.1 complements Table 1 in the main text, reporting all base models and all ensemble combinations across the six datasets and four ranking metrics (NDCG@10, MRR, AUC, Recall@10).

This fuller comparison confirms the main-text findings: as in Table 1, the three-way ensemble of text, categorical, and interactions achieves the highest average score on every dataset, and no single base representation dominates across datasets.

E.2 Cold-start vs. warm-start users

Having varied the evaluation metric, we next check that the all-users comparison is not driven by a single user regime: Figures E.1, E.2, E.3, and E.4 disaggregate test users into cold-start ($k < 10$ profile reviews) and warm-start ($k > 40$) cohorts, reporting NDCG@10, MRR, Recall@10, and ROC AUC, respectively. The qualitative picture carries over to both regimes: the three-way ensemble remains strongest overall, and no single base representation dominates across datasets in either cohort. The popularity baseline is the exception, degrading sharply from cold to warm cohorts on every dataset, consistent with data-rich users drifting toward long-tail items (Figure C.1). Warm-start cohorts contain far fewer users, so their estimates carry visibly wider confidence intervals.

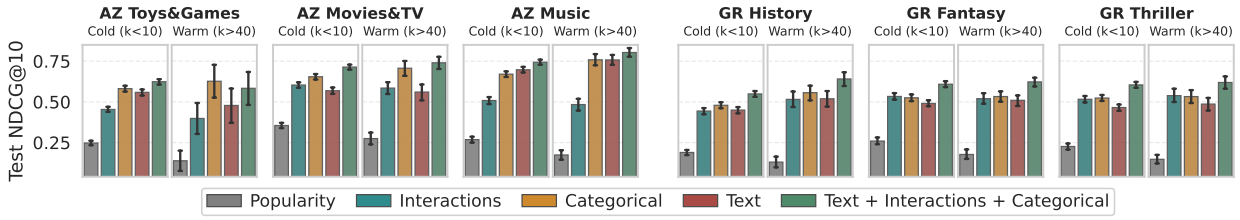


Figure E.1: [**Test**] NDCG@10 of the four base models and the three-way ensemble for cold-start ($k < 10$ profile reviews) and warm-start ($k > 40$) users. Bars show user means with bootstrapped 95% confidence intervals. Collaborative filtering (Interactions) is trained on at most 1M interactions.

E.3 Text vs. categorical across profile length

Figure E.5 compares text against categorical representations across per-user profile lengths (k).

Table E.1: User-wise test-set performance of base models and their ensembles on Goodreads (*GR*) and Amazon (*AZ*), across four ranking metrics (NDCG@10, MRR, AUC, Recall@10). Best overall in bold, best base model underlined; paired-Wilcoxon ties ($p > 0.05$) share the marker. The Interactions (CF) block reports four settings: 1M or 100K training interactions, each with hyperparameters tuned on validation NDCG@10 (as for the categorical MLP) or left at library defaults.

Preference representation (model)		AZ Toys&Games				AZ Movies&TV				AZ Music			
		NDCG	MRR	AUC	R@10	NDCG	MRR	AUC	R@10	NDCG	MRR	AUC	R@10
Text + Interactions + Categorical		**0.637**	**0.874**	**0.823**	**0.590**	**0.735**	**0.934**	**0.897**	**0.683**	**0.762**	**0.930**	**0.917**	**0.721**
Interactions + Categorical		0.616	0.847	0.810	0.575	0.723	0.915	0.893	**0.678**	0.705	0.877	0.892	0.673
Text + Categorical		0.618	0.843	0.816	0.577	0.705	0.921	0.882	0.654	**0.761**	**0.942**	0.914	0.714
Text + Interactions		0.583	0.838	0.790	0.535	0.668	0.911	0.848	0.610	0.707	0.894	0.887	0.667
Text (LLM)		0.567	0.818	0.771	0.522	0.584	0.822	0.800	0.540	0.713	0.922	0.880	0.662
Categorical *from Text* (MLP)		0.595	0.816	0.803	0.558	0.679	0.902	0.871	0.630	0.699	0.899	0.885	0.654
Interactions (CF)	*1M, tuned*	0.461	0.746	0.678	0.419	0.606	0.858	0.785	0.560	0.498	0.762	0.709	0.453
	1M, default	0.299	0.529	0.586	0.284	0.458	0.704	0.684	0.434	0.349	0.576	0.611	0.332
	100K, tuned	0.282	0.524	0.546	0.265	0.394	0.648	0.611	0.365	0.294	0.559	0.555	0.271
	100K, default	0.261	0.509	0.517	0.244	0.344	0.607	0.544	0.325	0.262	0.524	0.518	0.245
Popularity		0.233	0.422	0.559	0.231	0.330	0.494	0.707	0.344	0.234	0.394	0.550	0.242
Preference representation (model)		GR History				GR Fantasy				GR Thriller			
		NDCG	MRR	AUC	R@10	NDCG	MRR	AUC	R@10	NDCG	MRR	AUC	R@10
Text + Interactions + Categorical		**0.579**	**0.837**	**0.808**	**0.532**	**0.619**	**0.872**	**0.834**	**0.570**	**0.620**	**0.874**	**0.820**	**0.569**
Interactions + Categorical		0.566	0.819	0.795	0.521	0.605	0.859	0.822	0.557	0.607	0.859	0.809	0.557
Text + Categorical		0.544	0.815	0.792	0.502	0.567	0.824	0.813	0.525	0.570	0.821	0.796	0.526
Text + Interactions		0.523	0.774	0.773	0.487	0.585	**0.862**	0.797	0.531	0.575	0.840	0.781	0.524
Text (LLM)		0.464	0.742	0.734	0.433	0.497	0.778	0.747	0.455	0.479	0.763	0.720	0.436
Categorical *from Text* (MLP)		0.505	0.759	0.773	0.473	0.531	0.777	0.793	0.497	0.535	0.786	0.779	0.497
Interactions (CF)	*1M, tuned*	0.466	0.725	0.708	0.432	0.538	0.814	0.741	0.489	0.536	0.796	0.735	0.489
	1M, default	0.355	0.612	0.633	0.332	0.392	0.673	0.642	0.353	0.382	0.654	0.657	0.348
	100K, tuned	0.359	0.613	0.631	0.337	0.447	0.731	0.679	0.405	0.411	0.682	0.660	0.374
	100K, default	0.304	0.544	0.586	0.289	0.376	0.671	0.619	0.335	0.331	0.604	0.603	0.303
Popularity		0.180	0.342	0.497	0.183	0.236	0.413	0.541	0.236	0.206	0.384	0.489	0.204

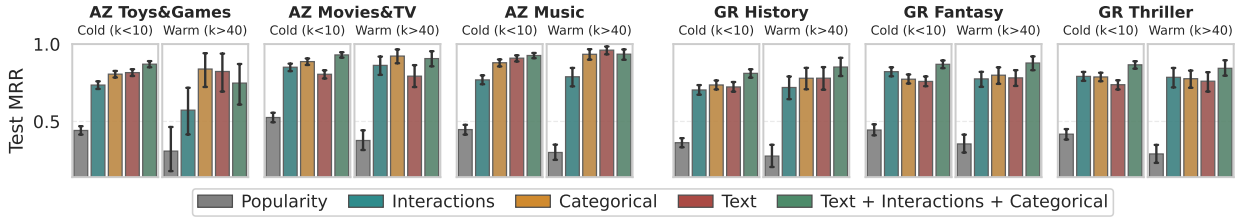


Figure E.2: [**Test**] MRR for cold-start ($k < 10$) and warm-start ($k > 40$) users, as in Figure E.1.

E.4 Robustness check: excluding evaluation users from categorical-model training

This subsection stress-tests the categorical model under a deliberately harder evaluation protocol than the one used in the main paper. In the main results, both trained scorers receive the same test-user data: each test user's profile interactions (never their held-out evaluation interactions). They differ only in how those interactions are used. Collaborative filtering fits each test user's latent factors on them at inference (fold-in; the item factors stay fixed), while the categorical

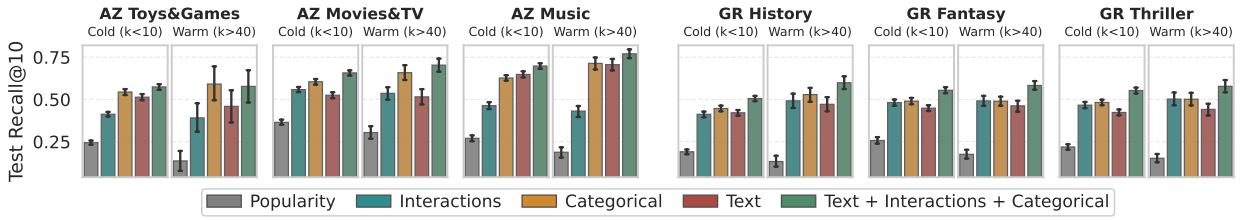


Figure E.3: [**Test**] Recall@10 for cold-start ($k < 10$) and warm-start ($k > 40$) users, as in Figure E.1.

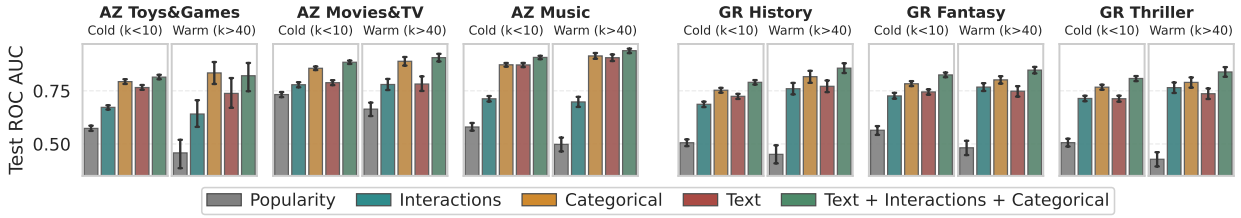


Figure E.4: [**Test**] ROC AUC for cold-start ($k < 10$) and warm-start ($k > 40$) users, as in Figure E.1.

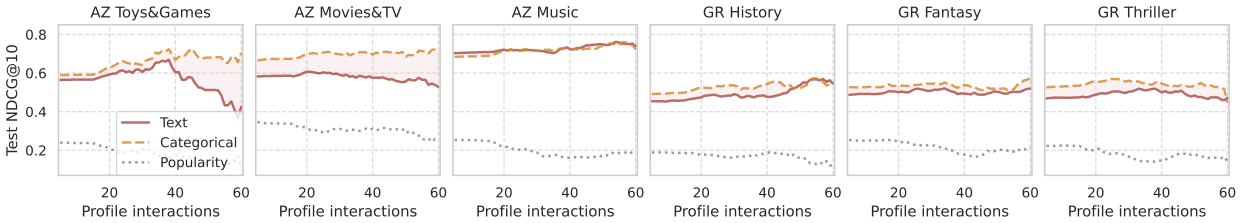


Figure E.5: Test-set NDCG@10 by profile length k (number of profile interactions/reviews); each k indexes a distinct user cohort: the test users whose profile length lies within ± 10 of k (moving average). Both representations derive from the same user-authored reviews; the categorical model is trained on dataset-specific data, while the text-based LLM is used zero-shot.

MLP includes them among the training pairs for its shared parameters. Neither model sees more data than the other; indeed, the fold-in is CF's only source of personalization, since without test users' profile interactions it would have nothing user-specific to go on and would fall back on an unpersonalized, popularity-based ranking. As an over-cautious robustness check, we nonetheless ask how the categorical model fares when no parameters are fitted on test users. We refit each dataset's selected configuration on the 1,000 validation users alone (a *validation-only* fit, $\approx$70% of the original training pairs) and evaluate on the same held-out test interactions. Test users' profiles are then used only to build each user's categorical representation at inference, just as the zero-shot text model reads them in its prompt. Both models thus still access test users' profile interactions at inference (CF folds in latent factors, the categorical model builds the tag vector); what changes is only the training sample for the shared parameters, now a few tens of thousands of validation-user interactions (cf. Table C.2) against CF's one million. Table E.2 groups models by whether they fit any parameters on test users' profiles; the validation-only model is still held to a standard CF never faces, since CF fits user-side factors on those profiles while the categorical model only reads them.

Even under this stricter standard, the Amazon comparisons hold. NDCG@10 drops by 0.06–0.07,

yet the validation-only fit stays ahead of tuned CF-1M on Toys&Games (0.54 vs. 0.46) and Music (0.64 vs. 0.50), and matches it on Movies&TV (0.61 vs. 0.61). On Goodreads the drop is larger (0.10–0.17 NDCG@10), and the validation-only fit falls below CF-1M on all three genres.

We tested whether this larger drop reflects tag coverage. Without clustering, each tag is its own feature, so the validation-only fit learns little about tags that no validation user carries. Such tags are common on Goodreads: only 39–46% of Goodreads test users' tag occurrences appear among validation users, against 64–67% on Amazon. We therefore repeated both fits with tags grouped into 100 clusters (Appendix D), all of which occur among validation users. The drop shrinks on every dataset and most on Goodreads, where it falls to 0.04–0.06 NDCG@10, close to Amazon's 0.02–0.05 (Table E.3). Most of the extra Goodreads drop thus reflects incomplete tag coverage in a training population of 1,000 users, not a failure of the categorical representation on new users. The drop that remains on every dataset also reflects the fact that validation users' tags come from GPT-5-mini and test users' tags from GPT-5.

The zero-shot text model is unaffected by excluding test users from training, as it fits no parameters on any user. Relative to that fixed reference, the validation-only fit falls below the text model on five of six datasets (all but Movies&TV), whereas the paper configuration matches it on average; the main-text finding that categorical features match the average performance of the free-form text representation thus applies to models fit on the available profile interactions, not to this deliberately restricted fit. We do not view this check as the right way to compare the models; any realistic deployment would fit the model on available profile interactions, as the main results do.

E.5 LLM querying costs

The text-based model uses GPT-5 in two steps: it first generates one natural-language preference profile per test user and then scores each user–item candidate pair. Table E.4 reports the observed API cost of the second step on the held-out test split. Across the six datasets, GPT-5 scored 297,600 user–item pairs for a total cost of $4,310.91, corresponding to an average of 1.45 cents per pair (95% CI: 1.25–1.65 cents, computed across dataset-level per-pair costs).

Profile generation is a one-time preprocessing cost per user rather than a per-candidate serving cost. In the saved GPT-5 profile-generation logs with complete API usage records covering 2,968 generated profiles, generating the natural-language user profile costs an average of $0.043 per user (95% CI: $0.035–$0.051), or 4.33 cents per user.

E.6 Cost and performance across LLMs

To see how these costs depend on the choice of LLM, we re-ran the text pipeline on AZ Movies&TV with five further API models in place of GPT-5 for generating test users' profiles and scoring their candidate pairs (Table E.5). Total API cost spans roughly two orders of magnitude, from $11 (GPT-6-luna) to $1,257 (Claude Opus 5.5). Zero-shot AUC ranges from 0.758 (GPT-5-mini) to 0.831 (Claude Opus 5.5); only Claude Opus 5.5 exceeds GPT-5 (0.800). Similar accuracy can come at very different cost: GPT-6.1-sol and GPT-6-astra reach nearly the same zero-shot AUC (0.789 and 0.790) at a fourfold difference in cost ($223 versus $942), and GPT-6-luna comes within 0.01 AUC of both for $11. Per-pair scoring accounts for 86–94% of each model's total. Changing the LLM moves the zero-shot scorer far more than the categorical MLP: across the six LLMs, the MLP's AUC and NDCG@10 vary by about 0.02, against 0.07 and 0.13 for zero-shot scoring. Stronger LLMs therefore mainly benefit the zero-shot scorer. On this dataset the categorical model scores higher than the

Table E.2: Over-cautious robustness check: the categorical (MLP) model evaluated with no parameters fitted on test users. User-wise test-set performance on Goodreads (*GR*) and Amazon (*AZ*) across four ranking metrics (NDCG@10, MRR, AUC, Recall@10); rows are grouped by whether any parameters are fitted on test users' profile interactions (never their held-out evaluation interactions). The *validation-only fit* refits the categorical MLP on the 1,000 validation users alone; like the zero-shot Text model, it reads test users' profiles only to build their representations at inference. In the paper configuration (*val. + test profiles*), the MLP's shared parameters also train on test users' profile interactions; tuned CF at 1M fits each test user's latent factors on them at inference (fold-in), its only source of personalization. The Δ row reports, per cell, the validation-only fit minus the val. + test fit. All categorical rows use each dataset's validation-selected hyperparameters; the Text and CF rows repeat the corresponding entries of Table E.1.

Preference representation (model)		AZ Toys&Games				AZ Movies&TV				AZ Music			
		NDCG	MRR	AUC	R@10	NDCG	MRR	AUC	R@10	NDCG	MRR	AUC	R@10
Not fitted on test users' profiles													
Text	*free-form (ZS LLM)*	0.567	0.818	0.771	0.522	0.584	0.822	0.800	0.540	0.713	0.922	0.880	0.662
Categorical *from Text* (MLP)	*validation-only fit*	0.538	0.760	0.765	0.507	0.612	0.866	0.835	0.565	0.638	0.856	0.850	0.599
	Δ *from val. + test*	−0.057	−0.056	−0.038	−0.051	−0.067	−0.036	−0.036	−0.065	−0.061	−0.043	−0.035	−0.055
Fitted on test users' profile interactions													
Categorical *from Text* (MLP)	*val. + test profiles (paper)*	0.595	0.816	0.803	0.558	0.679	0.902	0.871	0.630	0.699	0.899	0.885	0.654
Interactions (CF)	*1M, tuned; user factors folded in*	0.461	0.746	0.678	0.419	0.606	0.858	0.785	0.560	0.498	0.762	0.709	0.453
Preference representation (model)		GR History				GR Fantasy				GR Thriller			
		NDCG	MRR	AUC	R@10	NDCG	MRR	AUC	R@10	NDCG	MRR	AUC	R@10
Not fitted on test users' profiles													
Text	*free-form (ZS LLM)*	0.464	0.742	0.734	0.433	0.497	0.778	0.747	0.455	0.479	0.763	0.720	0.436
Categorical *from Text* (MLP)	*validation-only fit*	0.355	0.593	0.648	0.336	0.426	0.654	0.716	0.407	0.362	0.597	0.660	0.348
	Δ *from val. + test*	−0.150	−0.166	−0.125	−0.137	−0.105	−0.123	−0.077	−0.090	−0.173	−0.189	−0.119	−0.149
Fitted on test users' profile interactions													
Categorical *from Text* (MLP)	*val. + test profiles (paper)*	0.505	0.759	0.773	0.473	0.531	0.777	0.793	0.497	0.535	0.786	0.779	0.497
Interactions (CF)	*1M, tuned; user factors folded in*	0.466	0.725	0.708	0.432	0.538	0.814	0.741	0.489	0.536	0.796	0.735	0.489

Table E.3: Clustering tags removes most of the extra Goodreads drop. Change in test NDCG@10 (Δ) of the validation-only fit relative to the paper configuration (val. + test profiles), with the full tag vocabulary (the Δ row of Table E.2) and with tags grouped into 100 clusters.

Dataset	Δ, full vocabulary	Δ, 100 clusters	Reduction [95% CI]
AZ Toys&Games	−0.057	−0.023	0.034 [0.022, 0.046]
AZ Movies&TV	−0.067	−0.035	0.032 [0.020, 0.044]
AZ Music	−0.061	−0.046	0.016 [0.004, 0.027]
GR History	−0.150	−0.038	0.112 [0.096, 0.129]
GR Fantasy	−0.105	−0.042	0.063 [0.048, 0.078]
GR Thriller	−0.173	−0.057	0.115 [0.099, 0.132]

zero-shot scorer with every LLM, but this should not be read as a general advantage: Movies&TV is where the categorical model compares most favorably with the text model in the main results (0.679 versus 0.584 NDCG@10 with GPT-5), whereas on the other five datasets the categorical model's NDCG@10 differs from the text model's by −0.014 to +0.056 (Table 1). The MLP also makes no API calls at scoring time: its API cost is profile generation alone ($1–$128 for the test users).

Figure E.6 repeats the win-rate analysis of Figure 4 on AZ Movies&TV for each of these LLMs. Across LLMs and the four metrics, CF reaches parity at 0.3–4M training interactions (0.7–1.3M for

Table E.4: **GPT-5 user–item scoring costs on the test split.** Costs are computed from saved API usage metadata. The final row reports the total number of evaluated pairs and the mean cost per pair, with a 95% confidence interval across the six dataset-level costs.

Dataset	Pairs	Cost (USD)	Cents / pair
AZ Music	49,250	578.99	1.18
AZ Movies&TV	49,500	628.90	1.27
AZ Toys&Games	49,750	759.34	1.53
GR Fantasy	49,850	839.02	1.68
GR History	49,300	741.52	1.50
GR Thriller	49,950	763.13	1.53
Total / mean	297,600	4,310.91	1.45
95% CI			1.25–1.65

Table E.5: **API cost and test performance of the text pipeline with six LLMs on AZ Movies&TV.** Each row uses the listed LLM to generate test users' profiles (Profiles) and score their ≈ 50K candidate pairs zero-shot (Scoring); all else follows the main text (GPT-5; Tables 1 and E.4). Costs use API-reported token usage at list prices; rows are sorted by total cost. Costs reflect both per-token prices and reasoning-token usage: GPT-6-astra has higher per-token prices than Claude Opus 5.5 but uses far fewer reasoning tokens. Reasoning effort is the provider default except for GPT-6.1-sol (high). Metrics are user-wise (990–1,000 test users per run after content filtering; §C.5). The categorical MLP uses the validation-selected configuration (§D); its validation users' profiles come from GPT-5-mini (GPT-5 and GPT-5-mini rows) or GPT-6-luna (other rows).

	API cost (USD)			Free-form (ZS LLM)		Categorical (MLP)	
Model	Profiles	Scoring	Total	AUC	NDCG@10	AUC	NDCG@10
GPT-6-luna	1.31	9.23	10.54	0.781	0.560	0.863	0.667
GPT-5-mini	4.89	72.11	77.00	0.758	0.525	0.868	0.676
GPT-6.1-sol	28.33	194.21	222.54	0.789	0.579	0.863	0.673
GPT-5 (main text)	46.58	628.90	675.48	0.800	0.584	0.871	0.679
GPT-6-astra	128.30	813.37	941.67	0.790	0.580	0.857	0.665
Claude Opus 5.5	91.36	1,165.99	1,257.35	0.831	0.653	0.873	0.686

GPT-5). The crossover comes earliest for GPT-5-mini (0.4M on NDCG@10) and latest for Claude Opus 5.5 (2M on NDCG@10 and 4M on AUC).

E.7 Hyperparameter search

Both trained model families, collaborative filtering (Interactions) and the supervised MLP (Categorical), are tuned on validation NDCG@10. The selected configurations are then evaluated on test data and reported in the main text. The text representation involves no tuning: the LLM is used zero-shot with a fixed prompt. Tables E.6 and E.7 report the searched hyperparameter grids, and Figure E.7 the resulting collaborative-filtering validation landscape on all six datasets.

Collaborative filtering. The CF sweep is the largest in the paper: the $7 \times 11 \times 2 = 154$ hyperparameter combinations of Table E.6 are crossed with nine training-set size caps, giving 1,386

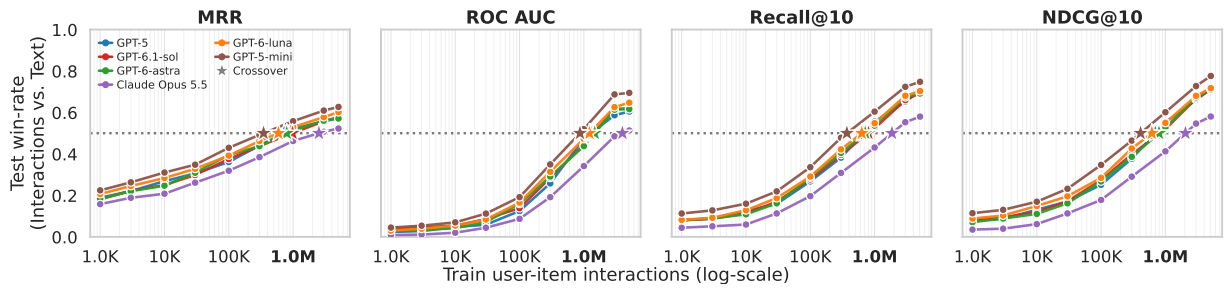


Figure E.6: **The CF–text crossover holds across LLMs.** Per-user test win-rate of Interactions (CF) over Text on AZ Movies&TV, as in Figure 4, with one line per LLM (Table E.5). The horizontal line marks parity (50% win-rate); stars mark each LLM's crossover. Win-rates use the test users common to the CF and LLM runs (990–1,000). Each LLM is compared against its own CF sweep.

Table E.6: **Collaborative filtering (implicit-feedback ALS): hyperparameter search space.** All 154 combinations are swept per dataset and size cap; caps above the full pool clamp to it.

Hyperparameter	**Searched values**
Latent factors	10; 25; 50; 100; 250; 500; 1,000
Confidence weight α	0.4; 1; 4; 10; 40; 100; 400; 1,000; 4,000; 10,000; 40,000
Regularization λ	0.01; 0.1
ALS iterations	100 (fixed)
Train interactions (cap)	1K; 3K; 10K; 30K; 100K; 300K; 1M; 3M; 10M

validation runs per dataset (≈8.3K in total). Figure E.7 shows the validation landscape over the two most impactful hyperparameters (the number of latent factors and the confidence weight α) when training on the full train pool, with each cell reporting the best score over the regularization axis. The landscape is smooth, with a single broad optimum per dataset, and is more sensitive to α than to the number of factors, markedly so on the Amazon datasets. On every dataset the selected optimum lies strictly inside the searched ranges: best factors fall in 50–250 (searched range 10–1,000) and best α in 100–4,000 (searched range 0.4–40,000), so the grid brackets the optima rather than truncating them. The same grid is swept at every training-set size; the main-text comparisons use the configurations selected at the corresponding caps (1M and 100K interactions).

Supervised categorical MLP. The model is fit in two phases. We first sweep the search space below on the validation split, fitting each configuration on the 1,000 validation users' profile interactions and scoring their held-out interactions, and keep the configuration with the best validation NDCG@10. That single configuration is then refit for the test evaluation on the profile interactions of all 2,000 evaluation users (Table C.2). The categorical model is deliberately simple, and tuning its architecture and regularization (search grid in Table E.7) helps only marginally, consistent with its small number of parameters and hyperparameters. On all six datasets, moving from a default architecture and regularization (a single hidden layer of 64 units with L2 penalty 10^{-4}) to the validation-selected configuration improves validation NDCG@10 by under 0.013, and changes *test* NDCG@10 by at most 0.018 (mean +0.010; because the configuration is selected on validation, on one dataset the default even edges out the selected one on the disjoint test split). Collaborative filtering, by contrast, gains +0.11 to +0.16 test NDCG@10 from the same default-

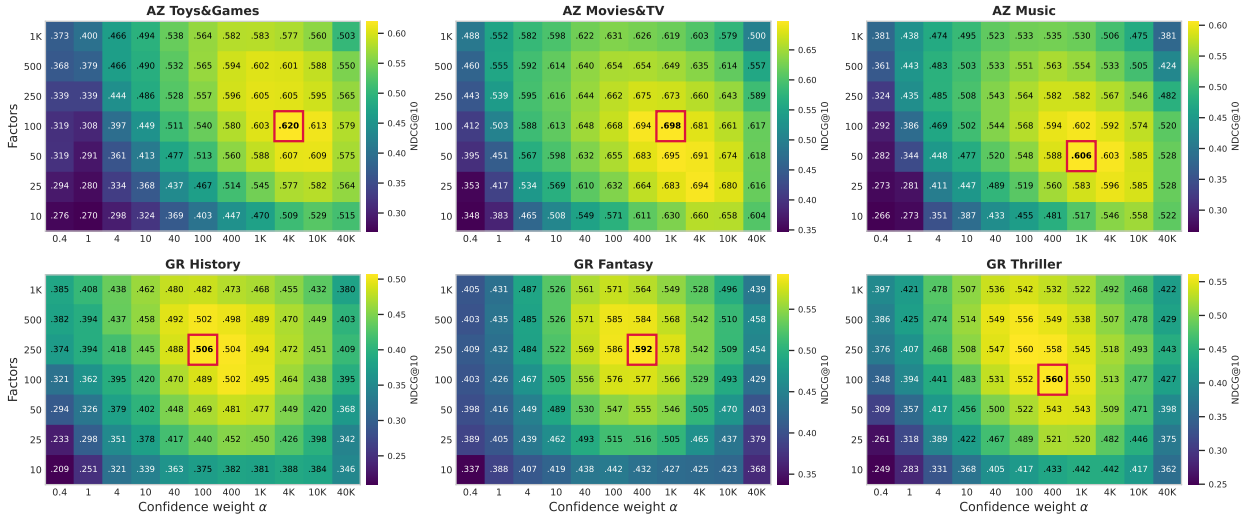


Figure E.7: **[All six datasets, Validation]** Collaborative-filtering hyperparameter search: validation NDCG@10 across the number of latent factors and the confidence weight α, trained on the full train pool. Each cell reports the best score over the regularization axis; the crimson box marks the per-dataset optimum; color scales are per panel.

Table E.7: **Supervised categorical MLP: hyperparameter search space.** The search is a union of two grids: the number of tag clusters is swept at two baseline architectures (hidden layers [64] and [128], at L2 10^{-4}), while the architecture grid (hidden layers $\times$ L2) is swept at 1,000 and 2,000 clusters and at the full vocabulary; both are crossed with the negative-sampling axes.

Hyperparameter	**Searched values**
Tag clusters	5; 10; 25; 50; 100; 250; 500; 1,000; 2,000; full vocabulary (no clustering)
Hidden layers	[32]; [64]; [64, 32]; [128]
L2 penalty	10^{-4}; 10^{-3}; 10^{-2}; 0.1
Negative sampling	uniform; popularity-weighted
Negatives per positive	5; 10; 20; 50

to-tuned move (mean +0.145), roughly an order of magnitude more, so hyperparameter search is far more consequential there (Table E.8). The axis that matters substantially is the tag-cluster granularity, which sets the feature representation rather than the model itself: finer granularities help, and the unclustered full vocabulary is at least as good as the best number of clusters on every dataset, so model selection retains it throughout.

Table E.8: **Default architecture/regularization vs. tuned hyperparameters (test NDCG@10).** For each *trained* model, test NDCG@10 under a default setting and under the configuration selected on validation NDCG@10, and their difference Δ. Categorical (MLP) default: a single hidden layer [64] with L2 penalty 10^{-4}, keeping the dataset's validation-selected full (unclustered) tag vocabulary and negative-sampling setting; tuned: the validation-selected architecture, regularization, and negative sampling. Interactions (CF) default: implicit-ALS library defaults (factors = 100, $\lambda = 0.01$, $\alpha = 1$, 15 iterations) at 1M training interactions; tuned: the validation-selected configuration at the same size. Because selection uses validation users, the tuned configuration can trail the default on the disjoint test users (negative Δ on AZ Music).

	Categorical (MLP)			Interactions (CF)		
Dataset	Default	Tuned	Δ	Default	Tuned	Δ
AZ Toys&Games	0.586	0.595	+0.009	0.299	0.461	+0.162
AZ Movies&TV	0.661	0.679	+0.018	0.458	0.606	+0.148
AZ Music	0.701	0.699	−0.002	0.349	0.498	+0.149
GR History	0.491	0.505	+0.014	0.355	0.466	+0.111
GR Fantasy	0.526	0.531	+0.005	0.392	0.538	+0.146
GR Thriller	0.521	0.535	+0.014	0.382	0.536	+0.154
Mean	0.581	0.591	+0.010	0.373	0.518	+0.145

F LLM prompting details

This section documents the literal prompt text used in our LLM-based pipelines, the controls that keep evaluation data out of prompts, why memorization of the public corpora is unlikely to explain the results, and which inputs can feed a text profile.

Leakage controls. To prevent test/validation information from leaking into prompts, all community-review blocks in item descriptions are drawn from the training split only. We also exclude the evaluation user's own review from the candidate item's community-review block.

F.1 Memorization of public data

Both the Goodreads and Amazon corpora are publicly available and predate the LLM's training, so GPT-5 may have seen some of these reviews during pretraining. Memorization of the source data is a separate concern from the direct leakage that the leakage controls above rule out. The scoring setup limits what such exposure could contribute. A scoring prompt consists of the generated profile, the candidate item's description, and train-split community reviews (Prompts F.6 and F.7); it carries no user identifiers or interaction dates, and the profile is an LLM-written summary rather than verbatim review text. The raw reviews behind the profile are thus never shown to the scorer, and a memorization account requires a chain of three links: from the generated summary back to the verbatim reviews it condenses, from those reviews to the identity of their author, and from that author to their held-out interactions. The quantitative results also leave little room for memorization. Item popularity, the item-level statistic a pretrained model is most likely to have absorbed, is neutralized by the popularity-weighted sampling of negatives, and the explicit popularity baseline trails the text model on every dataset (Table 1). Moreover, the categorical pipeline reaches the same performance as GPT-5 scoring while ranking items with an open-source embedding model

and a lightweight supervised scorer fit only on this study's data, forcing all user-side signal through 16.6 human-readable tags on average, generated in the same profile-generation step, a bottleneck where memorized item-specific content would be visible on inspection (§D). That performance survives this compression indicates the profile text encodes preference categories that generalize, rather than pointers to specific items. The worked example in §F.3 illustrates the mechanism qualitatively: the score gap between two same-genre candidates tracks the semantic content of the profile, not author-name or genre matching (Prompt F.8). What remains is the LLM's general knowledge of items and their reception, which is precisely the capability under study; collaborative filtering receives the analogous item-side signal through its one million training interactions.

F.2 Text profiles from activity traces

A textual profile requires either deliberate input from the user or organic user-authored text, such as the reviews we use here. Activity traces can also feed the summarizer; such traces carry some of the same drawbacks as interaction data, including platform-exposure bias, but they enter a text profile that the user can still read and amend, unlike the latent factors of a learned model. We do not evaluate profiles built from interaction traces alone, relying instead on actual user-authored text, though such profiles may be a viable option when text is scarce.

F.3 Prompt templates and worked examples

We follow a two-message format (system + user) whenever the API supports it, mirroring the OpenAI-style `messages=[{role: system}, {role: user}]` interface. To avoid repetition, prompts may reference reusable sub-templates via placeholders such as `{{book.description}}` and `{{review.description}}`. Unless otherwise stated, placeholders use double braces and dot notation (e.g., `{{user.top_authors}}`). All worked examples below follow a single Goodreads *Fantasy & Paranormal* user end-to-end: the free-text reading profile produced by the profile-generation prompt (Prompt F.5) is the same profile that is later consumed, as `{{user.text_profile}}`, by the user–book scoring prompt (Prompt F.8).

Prompt F.1: [Template] Book Description

```
- Title: {{ book.title }};
- Author(s): {{ ", ".join(a.name for a in book.authors) }};
- Published: {{ book.publication_month }} {{ book.publication_year }};
- Pages: {{ book.num_pages }};
- Genres: {{ ", ".join(g for g in book.genres) }};

- Summary:
{{ book.description }};

- Community User Reviews:
1. {{ community_review_1 }}
2. {{ community_review_2 }}
[...]
```

At scoring time each community review is rendered with the Book Review Description sub-template (Prompt F.2); we include up to five community reviews, drawn from the training split only,

excluding the evaluation user's own review, and ranked by helpful-vote count. Profile generation instead uses a *concise* variant of this description (`book.describe(concise=True, max_reviews=0)`): title, author(s), genres, and summary only, pipe-separated on a single line, with no publication metadata and no community reviews.

Prompt F.2: [Template] Book Review Description

```
Review: {{ review.text }} | Likes: {{ review.n_votes }} | Comments: {{
review.n_comments }} | User rating: {{ review.rating }} (out of 5 stars).
```

Each field is omitted when its value is missing, and the review text is truncated to at most 10,000 characters. The same sub-template renders both a user's own reviews (in profile generation) and a candidate book's community reviews (in scoring).

For user-profile generation, the reviewed-item descriptions do not include any community reviews: each item is rendered with `max_reviews=0`, so the profile-generation context contains item metadata/summary and the target user's own review only. Formally, let X denote the input text and Y the summary. Any information about a user's latent preferences Θ present in Y must originate in X, so by the data-processing inequality $I(\Theta; Y) \leq I(\Theta; X)$ (91). This statement holds provided the LLM's mapping $X \mapsto Y$ depends only on X.

Prompt F.3: [Template] User Reading Profile Generation (System message)

```
You are a book analysis expert who synthesizes a user's reading preferences based
on their complete history of book reviews and ratings.
- Carefully examine the user's past reviews and ratings to identify recurring
themes in what they enjoy and dislike in books.
- Look for specific patterns in favored genres, writing styles, narrative
structures, themes, character types, pacing, or any other distinctive elements.
- If the user predominantly reads a particular author, book series, or subgenre,
mention it explicitly.
- Write the response in a conversational style across three concise paragraphs in
the first person, as if the user themselves is summarizing their tastes. Describe
both what they like and what they dislike (if any). Avoid starting sentences
with phrases like "Based on my reading history" - instead, use direct and natural
expressions such as "I like..." or "I prefer...".
- At the end, on a separate line, include overarching tags, attributes, and key
characteristics of the user's reading preferences in a comma-separated list within
square brackets, as in the following format: "[<tag1>, <tag2>, ...]".
The write-up should be clear and natural, capturing the user's voice while being
informative. Avoid redundancy and focus on the most defining aspects of their
reading preferences. This write-up will later be used to make personalized book
recommendations.
```

Prompt F.4: [Template] User Reading Profile Generation (User message)

```
User has read {{ user.num_books }} books and gave an average rating of {{
user.avg_rating }} (out of 5).
```

```
Their most read genres are {{ ", ".join(user.top_genres) }}:
- {{ genre }}:  User read {{ genre.count }} book(s), and gave an average star
rating of {{ genre.avg_rating }}.
[...]

Their most read authors are {{ ", ".join(user.top_authors) }}:
- {{ author }}:  User read {{ author.count }} book(s), and gave an average star
rating of {{ author.avg_rating }}.
[...]

User's past books read and reviewed (from oldest to most recent read):
- {{ book_1.describe(concise=True, max_reviews=0) }} | User's {{ review_1 }}.
- {{ book_2.describe(concise=True, max_reviews=0) }} | User's {{ review_2 }}.
[...]
```

Prompt F.5 shows a worked example for a single Goodreads *Fantasy & Paranormal* user. The input is abridged to the statistics header and one of the user's 93 reviewed books; the output is the full generated profile, which becomes this user's `{{user.text_profile}}` for scoring.

Prompt F.5: [Example] User Reading Profile Generation (gpt-5, 93 profile reviews)

```
[ System message:  identical to the system-message template above.  ]

[ User message (abridged):  ]

User has read 93 books and gave an average rating of 2.9 (out of 5).

Their most read genres are fantasy, young-adult, fiction:
- Fantasy:  User read 93 book(s), and gave an average star rating of 2.9.
- Young-adult:  User read 29 book(s), and gave an average star rating of 3.0.
- Fiction:  User read 20 book(s), and gave an average star rating of 2.9.

Their most read authors are Holly Black, Sherwood Smith, Garth Nix:
- Holly Black:  User read 6 book(s), and gave an average star rating of 3.0.
- Sherwood Smith:  User read 6 book(s), and gave an average star rating of 2.7.
- Garth Nix:  User read 5 book(s), and gave an average star rating of 3.0.

User's past books read and reviewed (from oldest to most recent read):
- Title:  A Wizard of Mars (Young Wizards, #9) | Author(s):  Diane Duane | Genres:
fantasy, young-adult | Summary:  [...]  | User's Review:  My summary makes it sound
a lot more interesting than it actually is.  There are far too many characters and
not much plot.  The vast majority of this novel is just wizards standing around
deciding what to do, explaining what's happening, or rehashing what just happened.
For a story about the revival of the Martians, this is surprisingly boring.  |
Likes:  3 | Comments:  0 | User rating:  2 (out of 5).
[...]  (92 further books, from oldest to most recent)

[ Output (generated profile, this user's {{user.text_profile}}):  ]
```

I reach for fantasy in all its flavors---urban, YA, gaslamp---with a soft spot for stories where the city itself is a character and the magic has rules that surprise without cheating. I love inventive systems, tight internal logic, and big-hearted brains at work: Kate Griffin's London (Matthew Swift/Magicals Anonymous), Frances Hardinge's intricate, unsettling wonders, Kristin Cashore's Bitterblue, Felix Gilman's half-made West, Garth Nix's Old Kingdom-adjacent tales and clever shorts, Ben Aaronovitch's Folly cases, Holly Black's sharp-edged fae (especially The Darkest Part of the Forest), Seth Dickinson's geopolitical chess, Claire North's time loops, and a smart heist like Six of Crows. Give me libraries, archives, guilds, and the mechanics of how a world runs---politics, supply chains, and consequences---woven into momentum and payoff.

I'm happiest with competent, curious protagonists and ensembles that spark---banter, loyalty, and found family---where problems get solved through research, strategy, and creative use of magic rather than destiny alone. Sibling dynamics and feminist, queer-inclusive worlds delight me, as do clever climactic set pieces that reward setup. I enjoy short fiction when it's tight, imaginative, and (rare bliss) free of duds. Humor helps, but it should sharpen the characters and the world, not undercut them.

What loses me: bloated POVs that drain tension, repetitive recaps and monologues, catch-phrase crutches, and endings that fizzle into deus ex machina or abrupt epilogues. I have little patience for contrived love triangles, abusive or "alpha" romance masquerading as passion, or tonal whiplash where frothy romp meets torture. Grim for the sake of grim leaves me cold; so does twee middle-grade simplicity and hand-wavy logistics. I keep coming back to Holly Black's fae and Garth Nix's worlds; Sherwood Smith's universes fascinate me even when the execution frustrates. Most of all, I want purposeful pacing, a strong sense of place, and long arcs that actually resolve.

[fantasy-first, urban-fantasy, YA-fantasy, gaslamp-touch, city-as-character, inventive-magic-with-rules, political-intrigue, strong-worldbuilding, libraries-archives, heists, witty-banter, found-family, sibling-dynamics, feminist, queer-inclusive, clever-climaxes, short-story-friendly, dislikes-deus-ex-machina, dislikes-bloated-POV, anti-love-triangle, avoids-twee, balanced-humor, consistent-pacing, fae-mythology, London-vibes]

Prompt F.6: [Template] User–Book Interaction Scoring (System message)

You are a book recommendation expert. Specifically, you analyze a user's explicit reading preferences to provide a recommendation score for a given book. Your goal is to maximize user engagement. First, carefully examine the user's reading preferences, as provided in their own words. Identify key themes, genres, writing styles, and other elements that influence their enjoyment of books. Then, analyze the given book based on its summary, genre, themes, style, and other users' written reviews. Compare its characteristics to the user's stated preferences. Use your own knowledge of the given book and the book market to assess how well this book aligns with what the user enjoys. Think step-by-step and reason through your response. Finally, output a recommendation score for the given book between 0

```
and 10.  Your answer must strictly follow the format "[[RECOMMENDATION_SCORE]]",
where "[[0]]" means the user is unlikely to read the book at all and "[[10]]" means
the user is certain to read the book.
```

Prompt F.7: [Template] User–Book Interaction Scoring (User message)

```
User reading preferences:
{{ user.text_profile }}

Book description:
{{ book.describe(max_reviews=5) }}

How likely is it that the user will read this book next?
```

The book description is rendered with the Book Description sub-template (Prompt F.1), and `{{user.text_profile}}` is the profile produced by the generation prompt (Prompt F.5). The required `[[RECOMMENDATION_SCORE]]` output format (0–10) is specified only in the system message (Prompt F.6); a parsed score $s \in \{0, \ldots, 10\}$ is mapped to a read-probability $p = s/10$, then averaged over both candidate orderings to correct for option-order bias.

Prompt F.8 scores two candidate books for the *same* user as Prompt F.5; both are fantasy titles, yet the model assigns very different scores.

Prompt F.8: [Example] User–Book Interaction Scoring (same user; two candidate books)

```
Both candidates reuse the User-message template (Prompt F.7) and the system message
(Prompt F.6); only the book differs.  Here {{ user.text_profile }} is the profile
generated in Prompt F.5, and {{ book.describe(max_reviews=5) }} is set to each
candidate below.

=== Candidate A ===
- Title:  Foxglove Summer (Peter Grant, #5);
- Author(s):  Ben Aaronovitch;
- Published:  November 2014;
- Pages:  377;
- Genres:  fantasy, mystery, thriller, crime;
- Summary:
In the fifth of his bestselling series Ben Aaronovitch takes Peter Grant out of
[...]  London - to a small village in Herefordshire where the local police are
reluctant to admit that there might be a supernatural element to the disappearance
of some local children.  [...]  Travelling west with Beverley Brook, Peter soon
finds himself caught up in a deep mystery and having to tackle local cops and local
gods.  And what's more all the shops are closed by 4pm ...;
- Community User Reviews:
1.  Review:  Brilliant, well worth checking out the locations mentioned in the book
on the internet.  | Likes:  9 | Comments:  7 | User rating:  5 (out of 5).
2.  Review:  My only issue with this book is that it was too good, finished too
quickly and left me wanting more.  | Likes:  8 | Comments:  18 | User rating:  5
```

```
(out of 5).
[...]

Model output:  [[9]]


=== Candidate B ===
- Title:  Tunnels (Tunnels, #1);
- Author(s):  Roderick Gordon, Brian Williams;
- Published:  July 2007;
- Pages:  464;
- Genres:  fantasy, young-adult;
- Summary:
14-year-old Will Burrows has little in common with his strange, dysfunctional
family.  [...]  He unearths the unbelievable:  a subterranean society that time
forgot.  "The Colony" has existed unchanged for a century [...]  ruled by a
merciless overclass, the Styx.  Will must free his father---is he also about to
ignite a revolution?;
- Community User Reviews:
1.  Review:  My friend recommended this book to me [...]  now that I've started
this book, I'm having a hard time keeping my eyes open!  | Likes:  1 | Comments:  0
| User rating:  1 (out of 5).
2.  Review:  Not bad.  British YA authors are much different from American YA
authors [...]  I liked it, but I'm not sure I'd seek out the next installment.
| Likes:  0 | Comments:  0 | User rating:  3 (out of 5).
[...]

Model output:  [[2]]
```

The held-out labels record that this user later read *Foxglove Summer* but not *Tunnels*, and the model orders the two candidates accordingly, scoring them 9 versus 2 despite both being fantasy. *Foxglove Summer* matches preferences the profile states explicitly (urban fantasy in which the city is itself a character, a mystery/procedural backbone, found-family dynamics, and "magic [with] rules"), whereas *Tunnels* is penalized for traits the profile dislikes (slow, "bloated" pacing and inconsistent world-building), as echoed by its lukewarm community reviews. Because both books are fantasy and *Foxglove Summer*'s author is not among the user's most-read authors (Holly Black, Sherwood Smith, Garth Nix), the gap cannot be explained by genre or author-name matching alone; it requires reading the semantic content of the profile and the candidate's reviews.

Item tag retrieval (embedding model). The remaining template is not a generative prompt: it is the query passed to the `Qwen3-Embedding-8B` retrieval model (32) to place each item in the categorical tag space (§D). The item's Book Description (Prompt F.1, rendered with up to five sampled community reviews) is wrapped in the short instruction below; the resulting string is embedded and the item is assigned its five nearest tags by cosine similarity.

Prompt F.9: [Template] Tag Retrieval Query for Goodreads Datasets

```
Instruct: Given a book description and user reviews, retrieve relevant tags that
match the book.
Book description: {{book_description}}
```

The book-centric prompts and worked examples in this section, including this query, have Amazon counterparts that substitute "product" for "book", leaving the structure unchanged.