On the Convergent Validity of Offline Evaluation Designs for Recommender Systems

Sushobhan Parajuli
sp3886@drexel.edu
Dept. of Information Science
Drexel University
Philadelphia, PA, USA

Samira Vaez Barenji
svaez@drexel.edu
Dept. of Information Science
Drexel University
Philadelphia, PA, USA

Michael D. Ekstrand
mdekstrand@drexel.edu
Dept. of Information Science
Drexel University
Philadelphia, PA, USA

Abstract

Offline evaluation on historical interaction logs is the most common evaluation methodology for recommender systems. However, such evaluations depend on sparse, incomplete, or biased data, which raises concerns about whether commonly used evaluation setups reliably reflect true user preferences. In this work, we study how offline evaluation design choices affect the validity of recommender system comparisons. We evaluate a set of recommendation models across several evaluation setups that vary key factors such as data filtering thresholds and candidate set construction. To assess the validity of these configurations, we measure the correlation between model rankings obtained from conventional train–test splits on sparse interaction data and rankings from evaluations based on dense ground-truth user feedback. We use this agreement as an indication of their validity with respect to true user preferences. Our results show that the validity of sparse evaluation depends on the dataset and the specific dense evaluation targets, and that there is no uniformly best offline evaluation design.

CCS Concepts

• **Information systems** → **Recommender systems**; *Evaluation of retrieval results.*

Keywords

recommender systems, offline evaluation, measurement, validity, implicit feedback, sparsity, system ranking

ACM Reference Format:
Sushobhan Parajuli, Samira Vaez Barenji, and Michael D. Ekstrand. 2026. On the Convergent Validity of Offline Evaluation Designs for Recommender Systems. In *20th ACM Conference on Recommender Systems (RecSys '26), September 27-October 02, 2026, Minneapolis, MN, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3773078.3831818

1 Introduction

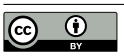
Offline evaluation based on historical interaction logs is the dominant approach for developing and comparing recommender systems (RS) in academic research, and serves as the first evaluation step in industrial settings where promising algorithms undergo further online trials. By splitting observed user–item interactions into training and test sets, researchers can efficiently evaluate models without requiring costly user studies or online experiments.

However, despite their widespread use, offline evaluation methods have significant limitations in how well they reflect true user preferences or estimate the likely online effectiveness of a proposed system. These limitations include *reliability* concerns, where results are heavily dependent on specific experimental setups or random seeds; *internal validity* concerns, caused by challenges such as extreme data sparsity and the biased nature of interaction data, where metrics may fail to capture what they are intended to measure; and *external validity* concerns, where conclusions do not necessarily transfer to other settings such as online deployment. Therefore, conclusions drawn from offline experiments may not faithfully reflect the underlying construct of interest, such as user satisfaction or system effectiveness.

In this paper, we examine challenges to the internal validity of offline top-N recommender system evaluation practices that arise from the extreme sparsity and biased observations in recommender system datasets. A key challenge is that we only observe feedback for a small subset of user–item pairs in these datasets, and that the missing interaction data is missing not at random (MNAR) [22, 23, 29]. These interactions cannot be reliably interpreted as either negative or unknown feedback, making it difficult to obtain valid measurements of model performance.

Offline evaluation results are further influenced by several experimental design choices, including data filtering, candidate set construction, and train–test splitting strategies. Prior work shows that these choices can substantially affect the ranking of recommendation models [11, 20, 24, 25], but it remains unclear which configurations most reliably reflect true user preferences. We therefore evaluate how well different evaluation setups on sparse data predict model rankings obtained from dense ground-truth feedback.

Using recently-released datasets that include relatively small but dense sets of user feedback, we assess the *convergent validity* between sparse top-N evaluations with sparse data and comparable evaluations using dense user feedback. Convergent validity, a concept from measurement theory [27, 33, 34], is a specific aspect of internal validity that is achieved when two different measurement procedures intended to capture the same construct produce consistent results. In our setting, this corresponds to comparing model rankings obtained from a commonly used, inexpensive measurement procedure (sparse offline evaluation) with rankings derived from a higher-quality but more costly alternative (dense ground-truth evaluation). Our study is guided by two key questions:



RecSys '26, Minneapolis, MN, USA

ACM ISBN 979-8-4007-2284-4/2026/09
https://doi.org/10.1145/3773078.3831818

RQ1 To what extent do model rankings obtained from sparse offline evaluation correlate with rankings derived from dense ground-truth evaluation?

RQ2 Do modifications to offline evaluation, such as sampling and data filtering, reduce the gap between these evaluations?

To address these questions, we evaluate a diverse set of recommendation models across a range of evaluation configurations, varying data filtering, candidate set construction, and list length. For each configuration, we evaluate the models on the sparse data and measure the correlation between those results and evaluation on the dense data. Our results suggest that the agreement between sparse and dense evaluation depends strongly on the dataset and evaluation setup, and that commonly observed differences between recommendation models may be driven as much by evaluation protocols as by model effectiveness.

2 Background and Related Work

Limitations of offline evaluation. A large body of prior work has examined the limitations of offline evaluation in RS, particularly its ability to reflect real-world performance and user preferences [13, 15, 16]. Rankings of algorithms derived from offline metrics often fail to generalize to user-facing settings, raising concerns about the external validity of standard evaluation protocols [2, 26]. Part of this gap is fundamental: even an ideal offline evaluation with perfect and complete or unbiased data may not fully capture interface effects, feedback loops, or long-term user satisfaction. That is, even a perfect offline evaluation may lack external validity. However, there is also a gap between a hypothetical ideal offline evaluation and what can be measured within the practical reality of sparse, biased interaction data and simplified evaluation protocols: offline evaluation in practice lacks internal validity with respect to what it purports to measure. We focus on the second component by examining how much evaluation design choices contribute to the gap between practical and ideal.

One major contributor to the internal validity gap is the nature of the data used for RS evaluation. Interaction data is typically sparse and missing not at random (MNAR), meaning that observed interactions are influenced by exposure and user behavior rather than random sampling [23, 29]. Therefore, missing interactions can carry useful information about user's taste. Steck [29] shows that treating missing interactions as unknown or randomly missing can introduce systematic bias in both training and evaluation. Later work proposed sampling techniques to attempt to approximate missing-at-random conditions or correct for exposure bias [5].

Evaluation as measurement. Measurement theory provides a number of validity and reliability properties for assessing the quality of empirical measurements [6, 18]. These properties provide a framework to more precisely characterize *how* internal or external validity may be violated. In this paper, we are particularly concerned with **convergent validity**: the extent to which multiple measurements intended to capture the same construct produce consistent results. Specifically, we compare offline evaluation results with sparse data to those obtained from dense ground-truth feedback to assess the extent to which these two measurements of model effectiveness converge on consistent answers.

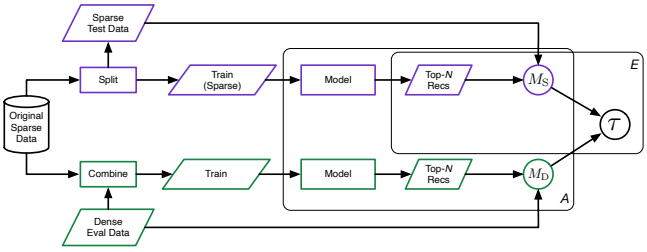


Figure 1: Diagram of experimental design for a single dataset and dense eval target. M_S and M_D are evaluation metrics over sparse and dense data, respectively, and the plates indicate repetition for A models and E sparse evaluation designs.

Evaluation design choices. Experimental design choices in offline evaluation can significantly affect the outcome of model comparisons. For example, different train–test splits, such as random, leave-one-out, or temporal splits, can lead to substantially different rankings of recommender systems [11, 24]. Sun [30] and Ji et al. [17] show that ignoring temporal ordering can introduce leakage and unrealistic evaluation scenarios. Verachtert et al. [31] suggest using an "optimal training window" of only recent data can drastically improve performance and change model rankings compared to using entire static datasets, as models otherwise learn patterns no longer aligned with current user interests; however, some forms of data pruning like k-core filtering can remove large portions of users and make results appear better by excluding cases on which algorithms perform worse [1].

Another offline evaluation decision of particular interest with respect to evaluation data selection bias is *candidate set sampling*: restricting the recommender to using a random subset of the items as candidates. While this has often been done for performance reasons [7, 19], it has also been proposed as a possible strategy to mitigate data biases [3, 9, 14]. Ihemelandu and Ekstrand [14] used simulation to find that weighting candidate sampling by item popularity may help reduce evaluation bias. Cañamares and Castells [4] found that minimum target sets can produce conclusions that differ from evaluation over the full candidate set, while neither minimal nor maximal target sets are uniformly optimal.

One of the persistent difficulties in evaluating evaluation designs is finding a reference point. Observing that two methods produce different results does not, on its own, tell us which method is more valid or reliable. For example, Krichene and Rendle [20] concluded that sampled candidate sets produce biased evaluations by comparing their results against full-set evaluations and finding significant differences in outcomes. However, if the sampling compensates for a bias in the original data, it may be more correct even though it differs. In this paper, we address this challenge by using smaller, harder-to-collect dense ground-truth data as a reference point for comparing evaluation designs over traditional sparse data.

3 Experimental Design

The core of our experimental design is to compare two evaluation setups applied to the same set of recommendation models, shown in Figure 1. In the **sparse setup**, we split the sparse data into training and test sets. We then train and evaluate the models using a typical

Table 1: Models employed.

Model	Variants
Popular	—
Bias	—
UserKNN	$k \in \{5, 20\}$, $k_{\min} \in \{1, 2, 5\}$
ItemKNN (implicit)	$k \in \{5, 20\}$, $k_{\min} \in \{1, 2, 5\}$
ItemKNN (explicit)	$k \in \{5, 20\}$, $k_{\min} \in \{1, 2, 5\}$
Implicit MF (ALS)	$d \in \{32, 64\}$
Biased MF (ALS)	$d \in \{32, 64\}$
BiasedSVD	$d \in \{32, 64\}$
NMF	$d \in \{32, 64\}$
BPR[a]	$d \in \{32, 64, 128, 256\}$
WARP[a]	$d \in \{32, 64, 128, 256\}$
Logistic MF[a]	$d \in \{32, 64, 128, 256\}$
Biased MF (SGD)[a]	$d \in \{32, 64\}$
fsSLIM	$k \in \{50, 100, 1000\}$
Total	45

[a]Implemented by LensKit FlexMF.

offline evaluation setup. In the **dense setup**, we train models on the entire sparse data and evaluate them against a dense ground-truth collected through an experimental intervention, in which users are exposed to many or all items to obtain much more complete preference data than is typically available. Each dataset therefore consists of two components: sparse data and dense ground-truth constructed through this intervention. The precise form of the dense ground-truth varies between datasets, and we describe it in more detail below.

We repeat the experiments on two different datasets (MovieLens and KuaiRec), a range of sparse evaluation design choices (candidate set construction, list length, and relevance threshold), and multiple dense evaluation targets. For each, we measure how well the sparse evaluation's ranking of recommendation models correlates with the ranking under dense evaluation using Kendall's τ. This allows us to assess the extent to which the sparse data available for offline recommender system evaluation is a valid proxy for dense ground truth (RQ1), and to identify which sparse evaluation design choices — candidate set construction, list length, and relevance threshold — best preserve that validity (RQ2).[1]

3.1 Datasets

MovieLens-32M. MovieLens-32M is a movie recommendation dataset containing 32 million explicit ratings from 200,948 users on a 0.5-5.0 scale [12]. The extended version proposed by Smucker and Chamani [28] augments this dataset with a dense set of relevance judgments collected from 51 active MovieLens users. These users were recruited and asked to assess recommended movies by indicating their interest in watching them, providing explicit preference judgments beyond their existing rating histories. To collect these judgments, recommendation lists from multiple algorithms were pooled together following standard information retrieval pooling methodology [21, 32]. Participants then evaluated the pooled items to produce a denser set of user–item relevance signals than what is available in sparse rating data.

The resulting dataset contains approximately 31.7 million ratings on the standard MovieLens 0.5-5.0 scale. We use this dataset as our sparse data for evaluation and as the training data for dense evaluation. For the dense ground-truth we use *interest.qrels* from the extension, which captures each participant's reported interest level in watching each movie on a 0–4 scale. We collapse the original 0–7 scale to 0–4 by merging the top ranking levels. For binary evaluation we threshold at interest level $r \geq 2$ (Interested) and $r \geq 3$ (Very Interested). For graded evaluation we use the 0–4 interest level as-is.

KuaiRec. KuaiRec is a short-video recommendation dataset collected from the Kuaishou platform [10]. The dataset contains user interactions based on watch behavior, where the primary feedback signal is the *watch ratio*, defined as the play duration divided by video duration. A watch ratio of 1.0 indicates the user watched the video at least once in full, and greater than 2.0 indicates the video was replayed. When using explicit models, we treat watch ratios as ratings.

KuaiRec provides a "big" matrix consisting of 7,176 users and 10,728 items at 16.3% density. We use this matrix as our sparse data for evaluation and as training data for dense evaluation. For dense ground-truth, we use the "small" matrix provided in KuaiRec, which contains 1,411 users and 3,327 items at 99.6% density. This matrix was collected through a controlled exposure process in which users were shown a large portion of the item catalog [10]. For binary evaluation we threshold at watch ratio $r \geq 0.85$ and $r \geq 2.0$. For graded evaluation we subtract 0.2 from the watch ratio and clip at zero, so videos watched less than 20% through receive zero gain.

3.2 Evaluation Setup

To generate the model rankings used in our comparison, we evaluated 45 recommendation models from LensKit 2026.1.0 [8], including both implicit and explicit feedback models. Table 1 lists the models and their variants. Explicit models were trained directly on the original feedback values. For implicit models, we additionally vary the minimum feedback value required for an interaction to be included in training — three cutoffs for MovieLens and four for KuaiRec. This yields 97 model configurations for MovieLens and 123 for KuaiRec, where each configuration corresponds to a distinct model and training threshold combination.

Sparse. In the sparse evaluation, we train models on the sparse data and evaluate them on a held-out test set from the same dataset. For MovieLens we use a temporal split with a fixed cutoff date so all training interactions occur before test interactions. The data in the KuaiRec matrices do not follow temporal order, so we use 5-fold cross-validation, holding out 5 interactions per user per fold.

We study three candidate set conditions: full candidates (all unseen items), U-1000 candidates (1,000 uniformly sample items), and P-1000 candidates (1,000 popularity-weighted sample items). We

[1]The code is available at https://doi.org/10.5281/zenodo.21549024

evaluate using both binary and graded relevance. For binary relevance, a test interaction is counted as relevant if the rating or watch ratio meets or exceeds a threshold. In the results tables, threshold rows reflect different ways of defining relevance at evaluation time, not different training configurations. We compute NDCG@k and DCG@k at ranking cutoffs $k \in \{10, 20, 100\}$.

Dense. In this setup, we train models on the sparse data and evaluate them against the dense ground-truth data described in Section 3.1. For MovieLens and KuaiRec, this corresponds to evaluation on the pooled user judgments (interest.qrels) and the small dense matrix, respectively. We evaluate using both binary and graded relevance and compute NDCG@k and DCG@k at ranking cutoffs $k \in \{10, 100\}$. Results at $k = 20$ were comparable to those at $k = 10$ and added little additional information, so we selected the two more widely separated cutoff values to represent distinct evaluation settings while keeping the result tables manageable.

4 Results

In this section, we examine the correlation between sparse and dense model rankings, and assess whether changes in evaluation design affect the gap between them. Our primary results are from Kendall's τ correlations between sparse and dense evaluation rankings of recommendation models. Each correlation is computed across the complete set of model configurations, including both explicit and implicit models. Tables 2 and 4 report Kendall's τ values.

4.1 MovieLens

Table 2 presents Kendall's τ values between sparse and dense model rankings for MovieLens. The dense target columns reflect two levels of user interest: Interested ($r \geq 2.0$) and Very Interested ($r \geq 3.0$), each at $k = 10$ and $k = 100$. We focus on Very Interested at $k = 10$ as the most practically meaningful target — identifying models that surface movies users are genuinely interested in watching within a short list — and use the other targets to contextualize the findings.

RQ1: Alignment between sparse and dense rankings. Sparse evaluation has positive correlation with dense evaluation under all designs in our selected dense target. For the dense target Very Interested and $k = 10$, the highest correlation is obtained with the sparse P-1000 candidate set at $k = 100$ and threshold $r \geq 3.0$, with $\tau = 0.29$. Figure 2 shows the relationship between model scores under these two evaluation configurations. The P-1000 candidate set generally produces higher τ values than the other candidate set designs, although the differences are small. For example, full candidates achieve $\tau = 0.26$ on the same sparse k and threshold. The correlation improves when the dense target is extended to $k = 100$, however the sparse design it improves on is a different one — full candidate set at sparse $k = 100$ and $r \geq 3.0$. This sparse evaluation design achieves $\tau = 0.43$, the highest value in the table. This improvement from $k = 10$ to $k = 100$ dense target may reflect that sparse evaluation is better suited to predicting which models place relevant items somewhere in a long list than which models surface them at the very top.

Table 3 shows the best models in dense evaluation and their rankings in sparse evaluation. Some of the top models that use implicit feedback in training — fsSLIM variants — rank consistently

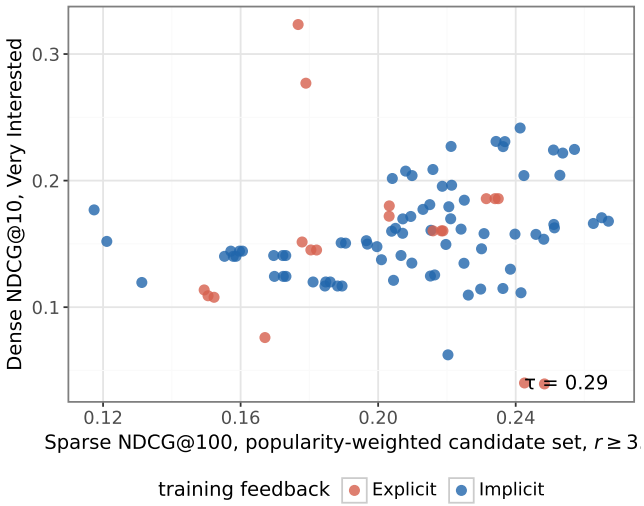


Figure 2: Scatter plot of NDCG scores under sparse evaluation (popularity-weighted candidate set, $k = 100$, $r \geq 3.0$) versus dense evaluation ($k = 10$, Very Interested) on MovieLens.

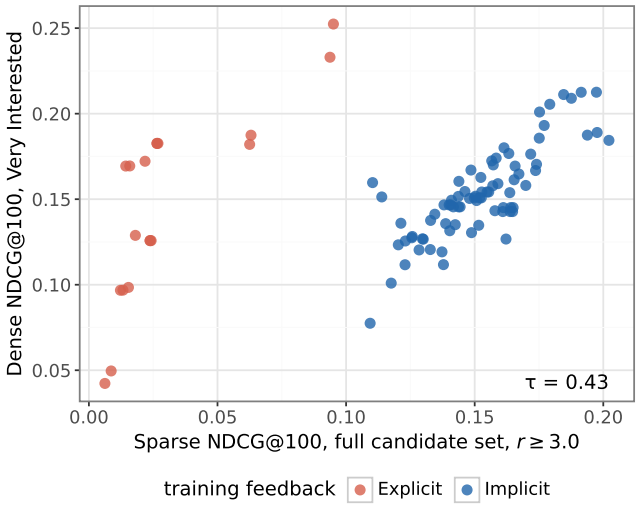


Figure 3: Scatter plot of NDCG scores under sparse evaluation (full candidate set, $k = 100$, $r \geq 3.0$) versus dense evaluation ($k = 100$, Very Interested) on MovieLens.

well under both sparse and dense evaluation. However, the best dense models overall that use explicit feedback — BiasedSVD-d64 and BiasedSVD-d32 — rank only 23^{rd} and 22^{nd} under their best sparse design, and rank 40^{th} and 38^{th} under the best-correlated sparse design (Full, $k = 100, r \geq 3.0$). When we compare the NDCG columns for these BiasedSVD variants, the metric values under dense evaluation are a lot higher than those under sparse evaluation. So, the BiasedSVD models perform significantly better on dense evaluation but are misranked on sparse evaluation. Similar ranking differences are visible for other models (see Figures 2 and 3). Models trained on implicit feedback tend to show more consistent

Table 2: Kendall's τ between sparse and dense evaluation model rankings by NDCG (MovieLens). Cells show τ with bootstrapped 95% confidence intervals. Bold indicates highest τ per dense target column.

			$k = 10$			$k = 100$		
Candidate set	k	Threshold	Interested	Very interested	Graded	Interested	Very interested	Graded
Full	10	≥ 3.0	0.11 (-0.06, 0.28)	0.21 (0.05, 0.36)	0.07 (-0.09, 0.24)	0.28 (0.11, 0.43)	0.38 (0.22, 0.52)	0.24 (0.08, 0.40)
		≥ 4.0	0.09 (-0.07, 0.26)	0.21 (0.05, 0.35)	0.05 (-0.12, 0.21)	0.26 (0.10, 0.41)	0.37 (0.22, 0.52)	0.22 (0.05, 0.37)
		Graded	0.10 (-0.06, 0.27)	0.22 (0.06, 0.36)	0.06 (-0.10, 0.23)	0.27 (0.11, 0.42)	0.38 (0.23, 0.53)	0.23 (0.07, 0.38)
	20	≥ 3.0	0.12 (-0.05, 0.29)	0.23 (0.06, 0.37)	0.08 (-0.08, 0.24)	0.28 (0.11, 0.44)	0.39 (0.23, 0.54)	**0.24 (0.07, 0.40)**
		≥ 4.0	0.10 (-0.06, 0.26)	0.22 (0.07, 0.36)	0.05 (-0.11, 0.21)	0.26 (0.10, 0.41)	0.38 (0.23, 0.52)	0.22 (0.06, 0.37)
		Graded	0.10 (-0.06, 0.27)	0.23 (0.07, 0.36)	0.06 (-0.11, 0.23)	0.27 (0.11, 0.42)	0.39 (0.23, 0.52)	0.23 (0.07, 0.38)
	100	≥ 3.0	**0.12 (-0.04, 0.28)**	0.26 (0.10, 0.40)	**0.08 (-0.08, 0.25)**	**0.29 (0.12, 0.44)**	**0.43 (0.26, 0.57)**	0.24 (0.07, 0.39)
		≥ 4.0	0.08 (-0.08, 0.24)	0.22 (0.07, 0.36)	0.03 (-0.13, 0.19)	0.24 (0.08, 0.39)	0.38 (0.22, 0.53)	0.19 (0.03, 0.34)
		Graded	0.09 (-0.07, 0.26)	0.23 (0.08, 0.37)	0.04 (-0.11, 0.21)	0.25 (0.09, 0.40)	0.40 (0.23, 0.54)	0.20 (0.04, 0.36)
U-1000	10	≥ 3.0	0.07 (-0.08, 0.23)	0.24 (0.08, 0.37)	0.01 (-0.14, 0.17)	0.18 (0.01, 0.34)	0.37 (0.21, 0.52)	0.13 (-0.03, 0.29)
		≥ 4.0	0.02 (-0.13, 0.17)	0.19 (0.04, 0.33)	-0.04 (-0.19, 0.11)	0.14 (-0.01, 0.30)	0.33 (0.17, 0.46)	0.09 (-0.06, 0.24)
		Graded	0.03 (-0.11, 0.19)	0.20 (0.06, 0.34)	-0.02 (-0.17, 0.14)	0.15 (0.00, 0.31)	0.34 (0.19, 0.48)	0.11 (-0.05, 0.26)
	20	≥ 3.0	0.07 (-0.08, 0.22)	0.23 (0.08, 0.37)	0.01 (-0.14, 0.17)	0.17 (0.00, 0.33)	0.37 (0.21, 0.51)	0.12 (-0.04, 0.28)
		≥ 4.0	0.00 (-0.14, 0.15)	0.18 (0.03, 0.32)	-0.05 (-0.20, 0.10)	0.12 (-0.04, 0.27)	0.31 (0.16, 0.45)	0.07 (-0.09, 0.22)
		Graded	0.02 (-0.13, 0.17)	0.20 (0.05, 0.33)	-0.04 (-0.18, 0.12)	0.13 (-0.03, 0.29)	0.33 (0.17, 0.47)	0.08 (-0.07, 0.24)
	100	≥ 3.0	0.06 (-0.09, 0.20)	0.23 (0.08, 0.36)	0.01 (-0.14, 0.16)	0.15 (-0.02, 0.31)	0.36 (0.20, 0.50)	0.11 (-0.05, 0.26)
		≥ 4.0	-0.02 (-0.16, 0.13)	0.17 (0.02, 0.30)	-0.07 (-0.21, 0.08)	0.08 (-0.08, 0.23)	0.29 (0.13, 0.43)	0.03 (-0.12, 0.19)
		Graded	0.00 (-0.14, 0.15)	0.19 (0.04, 0.32)	-0.05 (-0.20, 0.10)	0.10 (-0.06, 0.26)	0.31 (0.15, 0.45)	0.05 (-0.11, 0.20)
P-1000	10	≥ 3.0	0.06 (-0.09, 0.21)	0.28 (0.13, 0.41)	0.02 (-0.13, 0.17)	-0.02 (-0.17, 0.15)	0.20 (0.03, 0.35)	-0.05 (-0.20, 0.12)
		≥ 4.0	0.07 (-0.08, 0.19)	0.27 (0.14, 0.38)	0.05 (-0.09, 0.18)	0.05 (-0.11, 0.19)	0.26 (0.10, 0.38)	0.01 (-0.14, 0.16)
		Graded	0.07 (-0.07, 0.21)	0.28 (0.15, 0.39)	0.06 (-0.08, 0.20)	0.04 (-0.11, 0.19)	0.25 (0.09, 0.39)	0.01 (-0.14, 0.16)
	20	≥ 3.0	0.06 (-0.09, 0.21)	0.29 (0.13, 0.42)	0.02 (-0.12, 0.17)	-0.03 (-0.18, 0.13)	0.19 (0.03, 0.35)	-0.06 (-0.21, 0.10)
		≥ 4.0	0.06 (-0.08, 0.19)	0.27 (0.14, 0.37)	0.04 (-0.09, 0.18)	0.03 (-0.13, 0.17)	0.24 (0.09, 0.38)	-0.01 (-0.16, 0.13)
		Graded	0.07 (-0.07, 0.20)	0.28 (0.15, 0.39)	0.06 (-0.09, 0.19)	0.02 (-0.13, 0.16)	0.24 (0.08, 0.37)	-0.01 (-0.16, 0.13)
	100	≥ 3.0	0.05 (-0.10, 0.20)	**0.29 (0.14, 0.41)**	0.01 (-0.13, 0.16)	-0.05 (-0.20, 0.10)	0.18 (0.01, 0.33)	-0.08 (-0.22, 0.07)
		≥ 4.0	0.05 (-0.09, 0.18)	0.27 (0.13, 0.37)	0.04 (-0.10, 0.17)	-0.00 (-0.14, 0.14)	0.22 (0.07, 0.36)	-0.03 (-0.18, 0.10)
		Graded	0.05 (-0.09, 0.19)	0.27 (0.13, 0.38)	0.04 (-0.10, 0.18)	-0.02 (-0.16, 0.13)	0.21 (0.05, 0.34)	-0.05 (-0.19, 0.09)

Table 3: Top-5 models by dense NDCG@10 (k=10, Very Interested) and their best rank under the best-correlated sparse design and their best rank under any sparse design (MovieLens). The best dense model's sparse rank is in bold.

Model	Dense NDCG@10	Sparse NDCG (P-1000, $k = 100$, $r \geq 3.0$)	Rank (best corr.)	Best rank (any)	Best design
BiasedSVD-d64	**0.323**	**0.177**	**40**	**23**	Full, $k = 20$, $r \geq 4.0$
BiasedSVD-d32	0.277	0.179	38	22	Full, $k = 10$, Graded
fsSLIM-nn1000	0.242	0.267	1	1	U-1000, $k = 10$, $r \geq 3.0$
fsSLIM-nn50	0.231	0.263	3	3	U-1000, $k = 10$, $r \geq 3.0$
fsSLIM-nn100	0.231	0.265	2	1	U-1000, $k = 100$, $r \geq 3.0$

performance across sparse and dense evaluations, whereas models trained on explicit feedback show greater variation.

RQ2: Impact of evaluation design choices. Among the design choices we consider, candidate set has the largest visible effect on rank correlation, followed by relevance threshold and sparse k.

Against the dense target Very Interested and $k = 10$, the three candidate sets produce similar τ values — full candidate set at sparse $k = 10, r \geq 3.0$, achieves $\tau = 0.21$, U-1000 candidate set achieves $\tau = 0.24$, and P-1000 candidate set achieves $\tau = 0.28$. P-1000 candidate set is consistently better than the other designs at this dense target.

Against the Very Interested and $k = 100$ dense target, the candidate set effect becomes large and clear. Full candidate set at sparse $k = 100$, $r \geq 3.0$ achieves $\tau = 0.43$. U-1000 candidate set at the same sparse design drops to $\tau = 0.36$, and P-1000 candidate set drops further to $\tau = 0.18$. This reversal of P-1000 candidate set's relative performance between $k = 10$ and $k = 100$ dense targets suggests that popularity-weighted sampling may slightly favor models that

Table 4: Kendall's τ between sparse and dense evaluation model rankings by NDCG (KuaiRec). Cells show τ with bootstrapped 95% confidence intervals. Bold indicates highest τ per dense target column.

			$k = 10$			$k = 100$		
Candidate set	k	Threshold	Watched ≥85%	Watched 2×	Graded	Watched ≥85%	Watched 2×	Graded
Full	10	≥ 0.85	-0.36 (-0.45, -0.25)	-0.45 (-0.54, -0.35)	-0.48 (-0.56, -0.36)	-0.38 (-0.48, -0.27)	-0.49 (-0.57, -0.38)	-0.50 (-0.58, -0.38)
		≥ 2.0	-0.18 (-0.29, -0.05)	-0.25 (-0.36, -0.13)	-0.28 (-0.39, -0.15)	-0.24 (-0.35, -0.11)	-0.28 (-0.39, -0.15)	-0.31 (-0.43, -0.19)
		Graded	-0.36 (-0.45, -0.25)	-0.45 (-0.54, -0.35)	-0.48 (-0.57, -0.37)	-0.38 (-0.48, -0.27)	-0.49 (-0.57, -0.37)	-0.50 (-0.58, -0.38)
	20	≥ 0.85	-0.37 (-0.46, -0.27)	-0.47 (-0.55, -0.36)	-0.49 (-0.57, -0.39)	-0.39 (-0.48, -0.28)	-0.50 (-0.59, -0.39)	-0.51 (-0.59, -0.40)
		≥ 2.0	**-0.17 (-0.29, -0.05)**	-0.24 (-0.35, -0.11)	-0.27 (-0.38, -0.14)	**-0.24 (-0.35, -0.11)**	-0.27 (-0.38, -0.14)	-0.31 (-0.42, -0.18)
		Graded	-0.37 (-0.46, -0.27)	-0.47 (-0.55, -0.36)	-0.49 (-0.57, -0.39)	-0.40 (-0.48, -0.29)	-0.50 (-0.58, -0.39)	-0.51 (-0.59, -0.40)
	100	≥ 0.85	-0.42 (-0.50, -0.34)	-0.52 (-0.58, -0.43)	-0.54 (-0.61, -0.46)	-0.44 (-0.51, -0.36)	-0.55 (-0.62, -0.46)	-0.56 (-0.62, -0.47)
		≥ 2.0	-0.21 (-0.31, -0.10)	-0.24 (-0.34, -0.12)	-0.27 (-0.37, -0.15)	-0.27 (-0.38, -0.16)	-0.26 (-0.37, -0.14)	-0.30 (-0.41, -0.19)
		Graded	-0.43 (-0.51, -0.35)	-0.53 (-0.59, -0.45)	-0.55 (-0.62, -0.47)	-0.45 (-0.52, -0.38)	-0.56 (-0.63, -0.47)	-0.57 (-0.63, -0.49)
U-1000	10	≥ 0.85	-0.42 (-0.49, -0.34)	-0.51 (-0.58, -0.43)	-0.54 (-0.60, -0.45)	-0.43 (-0.50, -0.35)	-0.55 (-0.62, -0.46)	-0.55 (-0.61, -0.47)
		≥ 2.0	-0.19 (-0.29, -0.08)	-0.19 (-0.30, -0.09)	-0.22 (-0.33, -0.11)	-0.25 (-0.35, -0.14)	-0.21 (-0.32, -0.10)	-0.26 (-0.36, -0.14)
		Graded	-0.45 (-0.52, -0.37)	-0.54 (-0.60, -0.46)	-0.56 (-0.63, -0.48)	-0.46 (-0.53, -0.39)	-0.57 (-0.64, -0.49)	-0.58 (-0.64, -0.50)
	20	≥ 0.85	-0.43 (-0.51, -0.35)	-0.52 (-0.59, -0.44)	-0.55 (-0.62, -0.46)	-0.44 (-0.51, -0.37)	-0.55 (-0.62, -0.47)	-0.56 (-0.62, -0.48)
		≥ 2.0	-0.21 (-0.31, -0.11)	-0.21 (-0.31, -0.11)	-0.25 (-0.35, -0.14)	-0.27 (-0.37, -0.17)	-0.23 (-0.33, -0.13)	-0.28 (-0.38, -0.17)
		Graded	-0.46 (-0.53, -0.39)	-0.55 (-0.61, -0.47)	-0.58 (-0.64, -0.50)	-0.48 (-0.54, -0.41)	-0.59 (-0.65, -0.50)	-0.59 (-0.65, -0.52)
	100	≥ 0.85	-0.41 (-0.49, -0.34)	-0.50 (-0.57, -0.42)	-0.53 (-0.60, -0.45)	-0.42 (-0.49, -0.35)	-0.54 (-0.61, -0.45)	-0.54 (-0.60, -0.46)
		≥ 2.0	-0.23 (-0.33, -0.13)	-0.24 (-0.33, -0.13)	-0.27 (-0.36, -0.17)	-0.29 (-0.38, -0.18)	-0.26 (-0.35, -0.15)	-0.30 (-0.40, -0.19)
		Graded	-0.46 (-0.52, -0.38)	-0.54 (-0.60, -0.46)	-0.57 (-0.63, -0.50)	-0.47 (-0.53, -0.40)	-0.58 (-0.64, -0.50)	-0.59 (-0.64, -0.51)
P-1000	10	≥ 0.85	-0.43 (-0.51, -0.34)	-0.50 (-0.57, -0.41)	-0.53 (-0.60, -0.45)	-0.45 (-0.52, -0.37)	-0.53 (-0.60, -0.44)	-0.55 (-0.62, -0.47)
		≥ 2.0	-0.20 (-0.29, -0.09)	-0.19 (-0.28, -0.09)	-0.22 (-0.31, -0.12)	-0.25 (-0.33, -0.15)	-0.21 (-0.30, -0.10)	-0.25 (-0.34, -0.15)
		Graded	-0.45 (-0.53, -0.37)	-0.51 (-0.58, -0.43)	-0.55 (-0.62, -0.46)	-0.48 (-0.54, -0.40)	-0.54 (-0.61, -0.46)	-0.57 (-0.63, -0.49)
	20	≥ 0.85	-0.43 (-0.51, -0.35)	-0.49 (-0.56, -0.41)	-0.52 (-0.59, -0.44)	-0.45 (-0.52, -0.37)	-0.52 (-0.59, -0.44)	-0.54 (-0.61, -0.46)
		≥ 2.0	-0.20 (-0.29, -0.10)	**-0.19 (-0.28, -0.09)**	**-0.22 (-0.31, -0.12)**	-0.25 (-0.34, -0.15)	**-0.21 (-0.29, -0.11)**	**-0.25 (-0.34, -0.15)**
		Graded	-0.46 (-0.54, -0.38)	-0.51 (-0.57, -0.43)	-0.54 (-0.61, -0.47)	-0.49 (-0.55, -0.41)	-0.54 (-0.60, -0.46)	-0.57 (-0.63, -0.50)
	100	≥ 0.85	-0.42 (-0.51, -0.33)	-0.45 (-0.52, -0.36)	-0.49 (-0.56, -0.40)	-0.45 (-0.53, -0.36)	-0.47 (-0.54, -0.38)	-0.52 (-0.59, -0.43)
		≥ 2.0	-0.22 (-0.31, -0.12)	-0.21 (-0.30, -0.11)	-0.25 (-0.34, -0.14)	-0.27 (-0.36, -0.16)	-0.23 (-0.32, -0.13)	-0.28 (-0.36, -0.17)
		Graded	-0.46 (-0.54, -0.37)	-0.47 (-0.54, -0.39)	-0.51 (-0.58, -0.43)	-0.49 (-0.56, -0.41)	-0.50 (-0.56, -0.42)	-0.54 (-0.61, -0.46)

Table 5: Top-5 models by dense NDCG@10 (k=10, Watched 2×) and their rank under the best-correlated sparse design and their best rank under any sparse design (KuaiRec). The best dense model's sparse rank is in bold.

Model	Dense NDCG@10	Sparse NDCG (Full, $k = 20$, $r \geq 2.0$)	Rank (best corr.)	Best rank (any)	Best design
BiasedSVD-d32	**0.672**	**0.0**	**35**	**27**	P-1000, $k = 100$, $r \geq 2.0$
BiasedSVD-d64	0.665	0.001	32	26	P-1000, $k = 100$, $r \geq 2.0$
Bias	0.654	0.0	44	28	P-1000, $k = 10$, $r \geq 2.0$
ItemKNN-nnbrs5-min1-imp	0.649	0.059	18	18	Full, $k = 20$, $r \geq 2.0$
ItemKNN-nnbrs5-min2-imp	0.648	0.059	17	17	Full, $k = 20$, $r \geq 2.0$

perform well at short-list dense targets, but it does not improve alignment when the dense target is a longer list. Across all dense targets at $k = 100$ — Interested, Very Interested, and Graded — full candidate set designs produce consistently higher τ values. This suggests that candidate set choice has the most visible impact on model ranking correlation against dense evaluation.

Relevance threshold is a secondary factor. Graded sparse evaluation underperforms binary threshold $r \geq 3.0$, but slightly outperforms $r \geq 4.0$ against most dense targets. For example, full candidate set at sparse $k = 100$ with graded relevance achieve $\tau = 0.40$ against Very Interested and $k = 100$, compared to $\tau = 0.43$ with $r \geq 3.0$ and $\tau = 0.38$ with $r \geq 4.0$. This difference is small but consistent across full and U-1000 candidate sets. However, this pattern is mostly absent for P-1000 candidate set. Between binary thresholds, $r \geq 3.0$ outperforms $r \geq 4.0$ across full and U-1000 candidate sets. But, this pattern is reversed for P-1000 candidate set.

Sparse k has a small effect on rank correlation. τ values at $k = 10$, $k = 20$, and $k = 100$ are nearly identical within each candidate set and threshold combination. This suggests that the number of recommendations generated does not meaningfully affect how well sparse evaluation predicts dense rankings.

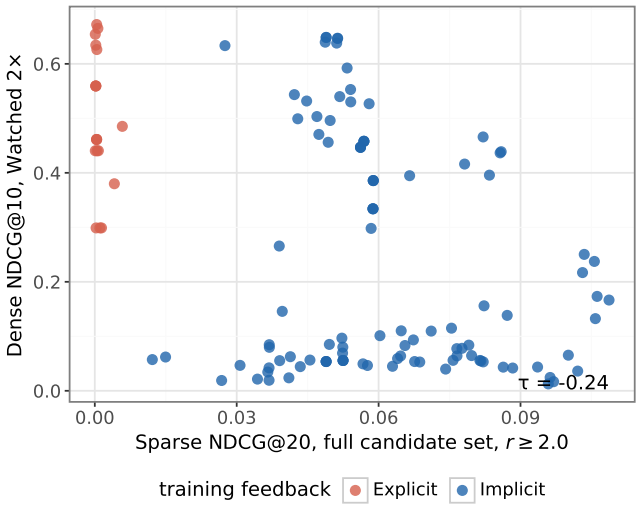


Figure 4: Scatter plot of NDCG scores under sparse evaluation (full candidate set, $k = 20$, $r \geq 2.0$) versus dense evaluation ($k = 10$, Watched 2x) on KuaiRec.

4.2 KuaiRec

While MovieLens showed weak but positive correlation between sparse and dense evaluations, our KuaiRec experiments find the evaluation designs to be negatively correlated (see Table 4). For this dataset, sparse evaluation inverts the dense ground-truth ranking regardless of any design choice. The two evaluation designs are incompatible for KuaiRec, and no evaluation design choice resolves this issue.

RQ1: Alignment between sparse and dense rankings. Against the dense target Watched 2x and $k = 10$, the least negative value is achieved by sparse P-1000 candidate set at $k = 20, r \geq 2.0$: $\tau = -0.19$. Every other design produces more negative τ. Against the Watched $\geq 85\%$ and $k = 10$, the least negative τ is at sparse full candidate set, $k = 20, r \geq 2.0$. Against the dense target Graded, τ values are most extreme — ranging to -0.57 — which indicates that graded relevance signals amplify the inversion most severely. Even under the most favorable sparse design, the top-5 dense models rank between 17^{th} and 28^{th} (see Table 5). Under the best-correlated sparse design (Full, $k = 20$, $r \geq 2.0$), ranks worsen further, ranging from 17^{th} to 44^{th}. Overall, sparse evaluation fails to reflect true model performance, as further illustrated in Figure 4, where models trained with explicit feedback perform poorly.

These findings from KuaiRec data show that, for this dataset, sparse evaluation can produce model rankings that are effectively inverted relative to dense ground-truth rankings.

RQ2: Impact of evaluation design choices. Among the design choices we consider, relevance threshold has the largest effect. For the Watched 2x and $k = 10$, sparse full candidates at $k = 20$ achieve $\tau = -0.24$ at $r \geq 2.0$. However, for other sparse threshold designs under the same candidate set and k the τ decreases to -0.47. Sparse $k = 20$ produces the least negative τ values across all candidate sets and dense targets.

Candidate set construction has a small effect on rank correlation of the models. Sparse P-1000 candidate set at $k = 20$ and $r \geq 2.0$ achieve $\tau = -0.19$ against the Watched 2x and $k = 10$, compared to $\tau = -0.24$ for sparse full candidate set and $\tau = -0.21$ for sparse U-1000 candidate set. However, given the confidence interval overlap we do not draw strong conclusions that changes in candidate set construction significantly impact model rank correlation.

5 Discussion and Limitations

Our results show that the correlation between sparse and dense evaluation varies by dataset and evaluation target. For MovieLens, there is positive correlation under most binary evaluation designs, while graded designs show lower correlations, with more negative values compared to binary evaluation designs. For KuaiRec, sparse evaluation inverts dense model rankings across all design choices, producing mostly negative correlations.

The contrast between the mostly positive MovieLens correlations and the consistently negative KuaiRec correlations could be partially attributed to their underlying characteristics, although we did not isolate the individual effects of such factors in our analysis. First, watch ratios in KuaiRec are implicit signals, while ratings in MovieLens are explicit signals. As a result, explicit models may have performed worse in the sparse evaluation design, where watch ratios are treated as ratings. Second, the KuaiRec dataset does not preserve temporal order, and the experiments rely on cross-validation as a non-temporal split, which may not reflect realistic recommendation scenarios. KuaiRec also contains fewer users than MovieLens, which may further affect the stability of evaluation results. For MovieLens, the positive correlations may be the effect of pooling bias and k-core filtering in the extended version that we used, which emphasize popular items and increase overlap between sparse and dense evaluation; we leave further analysis of this effect for future work.

We also observe that design modifications to sparse evaluation can affect alignment with dense evaluation. For MovieLens, full candidate sets and binary thresholds tend to produce higher correlations than sampled candidate sets and graded evaluation. These findings are consistent with prior work [20], in that sparse evaluation with full candidate sets is better correlated with dense evaluation than evaluation with uniformly sampled candidate sets. Popularity-weighted sampling can improve correlation for some dense evaluation targets, but that improvement is not consistent across evaluation targets. Our results are mostly consistent with their recommendation to avoid sampling, but confirm their results with a stronger reference point instead of treating the full-set sparse evaluations as correct, and show that popularity-weighted sampling can provide benefit in some situations.

The MovieLens results also suggest that sparse evaluation can be useful under the appropriate design (full candidates, binary threshold), but the gap between short and long list length suggests that sparse evaluation is more favorable at larger cutoff values. For KuaiRec, no design modifications on sparse evaluation produced positive correlation with the dense evaluation.

Overall, these observations suggest that commonly observed differences between recommendation models may be driven as much by evaluation design as by model effectiveness, and that models

identified as strongest under sparse evaluation are not necessarily those that best reflect user preferences as measured by dense evaluation. The variation in results across dense evaluation targets further suggests that sparse offline evaluation does not consistently capture a single effectiveness construct, and that different targets may require different evaluation designs, rather than there being a uniformly best offline evaluation setup.

6 Conclusion

We studied to what extent sparse offline evaluation correlates with dense ground-truth evaluation, and whether modifications to evaluation design can reduce the gap. Across two datasets and different evaluation designs, our results show that the relationship depends on the dataset and specific evaluation targets, with correlations ranging from weakly positive to strongly negative. Sparse evaluation mostly shows positive correlation with dense evaluation for MovieLens; however, it nearly reverses the model ranking from dense evaluation for KuaiRec. These results suggest that the validity of sparse evaluation is not consistent across settings, and that observed differences between recommendation models may be influenced by evaluation design as much as by model effectiveness.

Acknowledgments

This work was supported by the National Science Foundation under Grant IIS 24-09199.

References

[1] Joeran Beel and Victor Brunel. 2019. Data Pruning in Recommender Systems Research: Best-Practice or Malpractice?. In *Proceedings of the 13th ACM Conference on Recommender Systems (RecSys)*.

[2] Joeran Beel and Stefan Langer. 2015. A Comparison of Offline Evaluations, Online Evaluations, and User Studies in the Context of Research-Paper Recommender Systems. In *Research and Advanced Technology for Digital Libraries*, Sarantos Kapidakis, Cezary Mazurek, and Marcin Werla (Eds.). Vol. 9316. Springer International Publishing, Cham, 153–168. doi:10.1007/978-3-319-24592-8_12

[3] Alejandro Bellogín, Pablo Castells, and Iván Cantador. 2017. Statistical Biases in Information Retrieval Metrics for Recommender Systems. *Information Retrieval Journal* 20, 6 (Dec. 2017), 606–634. doi:10.1007/s10791-017-9312-z

[4] Rocío Cañamares and Pablo Castells. 2020. On Target Item Sampling in Offline Recommender System Evaluation. In *RecSys '20*. ACM, New York, NY, USA, 259–268. doi:10.1145/3383313.3412259

[5] Diego Carraro and Derek Bridge. 2022. A Sampling Approach to Debiasing the Offline Evaluation of Recommender Systems. *Journal of Intelligent Information Systems* 58, 2 (April 2022), 311–336. doi:10.1007/s10844-021-00651-y

[6] David A. Cook and Thomas J. Beckman. 2006. Current Concepts in Validity and Reliability for Psychometric Instruments: Theory and Application. *The American Journal of Medicine* 119, 2 (Feb. 2006), 166.e7–166.e16. doi:10.1016/j.amjmed.2005.10.036

[7] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of Recommender Algorithms on Top-N Recommendation Tasks. In *Proceedings of the Fourth ACM Conference on Recommender Systems*. ACM, Barcelona, Spain, 39–46. doi:10.1145/1864708.1864721

[8] Michael D Ekstrand. 2020. LensKit for Python: Next-generation software for recommender systems experiments. In *Proceedings of the 29th ACM international conference on information & knowledge management*. 2999–3006.

[9] Michael D Ekstrand and Vaibhav Mahant. 2017. Sturgeon and the Cool Kids: Problems with Top-N Recommender Evaluation. In *Proceedings of the 30th Florida Artificial Intelligence Research Society Conference (FLAIRS 30)*. AAAI Press.

[10] Chongming Gao, Shijun Li, Wenqiang Lei, Jiawei Chen, Biao Li, Peng Jiang, Xiangnan He, Jiaxin Mao, and Tat-Seng Chua. 2022. KuaiRec: A Fully-observed Dataset and Insights for Evaluating Recommender Systems. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. ACM, Atlanta, GA, USA, 540–550. doi:10.1145/3511808.3557220

[11] Danil Gusak, Anna Volodkevich, Anton Klenitskiy, Alexey Vasilev, and Evgeny Frolov. 2025. Time to Split: Exploring Data Splitting Strategies for Offline Evaluation of Sequential Recommenders. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*. ACM, Prague, Czech Republic, 874–883. doi:10.1145/3705328.3748164

[12] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4 (Dec. 2015), 19:1–19:19. doi:10.1145/2827872

[13] Balázs Hidasi and Ádám Tibor Czapp. 2023. Widespread Flaws in Offline Evaluation of Recommender Systems. In *Proceedings of the 17th ACM Conference on Recommender Systems*. ACM, Singapore, Singapore, 848–855. doi:10.1145/3604915.3608839

[14] Ngozi Ihemelandu and Michael D. Ekstrand. 2023. Candidate Set Sampling for Evaluating Top-N Recommendation. In *Proceedings of the 22nd IEEE/WIC International Conference on Web Intelligence and Intelligent Agent Technology*. 88–94. arXiv:2309.11723 [cs.IR] doi:10.1109/WI-IAT59888.2023.00018

[15] Amir H. Jadidinejad, Craig Macdonald, and Iadh Ounis. 2022. The Simpson's Paradox in the Offline Evaluation of Recommendation Systems. *ACM Transactions on Information Systems* 40, 1 (Jan. 2022), 1–22. doi:10.1145/3458509

[16] Olivier Jeunen. 2019. Revisiting Offline Evaluation for Implicit-Feedback Recommender Systems. In *Proceedings of the 13th ACM Conference on Recommender Systems*. ACM, Copenhagen, Denmark, 596–600. doi:10.1145/3298689.3347069

[17] Yitong Ji, Aixin Sun, Jie Zhang, and Chenliang Li. 2023. A Critical Study on Data Leakage in Recommender System Offline Evaluation. *ACM Transactions on Information Systems* 41, 3 (July 2023), 1–27. doi:10.1145/3569930

[18] Carole L. Kimberlin and Almut G. Winterstein. 2008. Validity and Reliability of Measurement Instruments Used in Research. *American Journal of Health-System Pharmacy* 65, 23 (Dec. 2008), 2276–2284. doi:10.2146/ajhp070364

[19] Yehuda Koren. 2008. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, Las Vegas, Nevada, USA, 426–434. doi:10.1145/1401890.1401944

[20] Walid Krichene and Steffen Rendle. 2020. On Sampled Metrics for Item Recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, New York, NY, USA, 1748–1757. doi:10.1145/3394486.3403226

[21] Xiaolu Lu, Alistair Moffat, and J. Shane Culpepper. 2016. The Effect of Pooling and Evaluation Depth on IR Metrics. *Information Retrieval Journal* 19, 4 (Aug. 2016), 416–445. doi:10.1007/s10791-016-9282-6

[22] Benjamin Marlin, Richard S. Zemel, Sam Roweis, and Malcolm Slaney. 2012. Collaborative Filtering and the Missing at Random Assumption. doi:10.48550/arXiv.1206.5267

[23] Benjamin M. Marlin and Richard S. Zemel. 2009. Collaborative Prediction and Ranking with Non-Random Missing Data. In *Proceedings of the Third ACM Conference on Recommender Systems*. ACM, New York, New York, USA, 5–12. doi:10.1145/1639714.1639717

[24] Zaiqiao Meng, Richard McCreadie, Craig Macdonald, and Iadh Ounis. 2020. Exploring Data Splitting Strategies for the Evaluation of Recommendation Models. In *Fourteenth ACM Conference on Recommender Systems*. ACM, Virtual Event, Brazil, 681–686. doi:10.1145/3383313.3418479

[25] Bruno L. Pereira, Alan Said, and Rodrygo L. T. Santos. 2025. On the Reliability of Sampling Strategies in Offline Recommender Evaluation. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems (RecSys '25)*. ACM, New York, NY, USA, 360–369. doi:10.1145/3705328.3748086

[26] Marco Rossetti, Fabio Stella, and Markus Zanker. 2016. Contrasting Offline and Online Results When Evaluating Recommendation Algorithms. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, Boston, Massachusetts, USA, 31–34. doi:10.1145/2959100.2959176

[27] Olawale Salaudeen, Anka Reuel, Ahmed Ahmed, Suhana Bedi, Zachary Robertson, Sudharsan Sundar, Ben Domingue, Angelina Wang, and Sanmi Koyejo. 2025. Measurement to Meaning: A Validity-Centered Framework for AI Evaluation. doi:10.48550/arXiv.2505.10573

[28] Mark D. Smucker and Houmaan Chamani. 2025. Extending MovieLens-32M to Provide New Evaluation Objectives. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Padua, Italy, 3520–3529. doi:10.1145/3726302.3730328

[29] Harald Steck. 2010. Training and Testing of Recommender Systems on Data Missing Not at Random. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, Washington DC USA, 713–722. doi:10.1145/1835804.1835895

[30] Aixin Sun. 2023. Take a Fresh Look at Recommender Systems from an Evaluation Standpoint. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Taipei Taiwan, 2629–2638. doi:10.1145/3539618.3591931

[31] Robin Verachtert, Lien Michiels, and Bart Goethals. 2022. Are We Forgetting Something? Correctly Evaluate a Recommender System with an Optimal Training Window. In *Proceedings of the Perspectives on the Evaluation of Recommender Systems Workshop 2022*.

[32] Ellen M Voorhees, Donna K Harman, et al. 2005. *TREC: Experiment and evaluation in information retrieval*. Vol. 63. MIT press Cambridge.

[33] Hanna Wallach, Meera Desai, A. Feder Cooper, Angelina Wang, Chad Atalla, Solon Barocas, Su Lin Blodgett, Alexandra Chouldechova, Emily Corvi, P. Alex Dow, Jean Garcia-Gathright, Alexandra Olteanu, Nicholas Pangakis, Stefanie Reed, Emily Sheng, Dan Vann, Jennifer Wortman Vaughan, Matthew Vogel, Hannah Washington, and Abigail Z. Jacobs. 2025. Position: Evaluating Generative AI Systems Is a Social Science Measurement Challenge. https://arxiv.org/abs/2502.00561v2.

[34] Xiting Wang, Liming Jiang, José Hernández-Orallo, David Stillwell, Shiqiang Chen, Luning Sun, Fang Luo, and Xing Xie. 2026. Evaluating General-Purpose AI with Psychometrics. *Commun. ACM* (April 2026), 3769688. doi:10.1145/3769688