Training seeds and model-selection stability in recommender-system evaluation

Juan Manuel Rodriguez
jmro@cs.aau.dk
Aalborg University
Aalborg, Denmark

Oleg Lesota
oleg.lesota@jku.at
Johannes Kepler University Linz
Linz, Austria

Antonela Tommasel
antonela.tommasel@jku.at
Johannes Kepler University Linz
Linz, Austria
ISISTAN, CONICET-UNCPBA
Tandil, Argentina

Abstract

Recommender-system experiments often rely on a single random training seed, assuming that run-to-run stochasticity has limited impact on evaluation conclusions. This assumption is risky, as a training seed may influence several algorithm-dependent mechanisms, including parameter initialization, mini-batch ordering, dropout, masking, latent sampling, and training-time negative sampling. We examine this assumption by fixing the data partition and varying the training seed across hyperparameter configurations. We analyze seed effects at three levels: user-level metric sensitivity, validation-based model selection and recommendation-list agreement. Results show that seed variation is often detectable. Its impact depends on whether configurations are clearly separated, whether validation results transfer to test, and whether similar scores lead to similar top-k lists. Findings suggest that reporting single-seed results can overstate the stability of recommender system evaluation, and that training seeds should be treated as part of the evaluation protocol rather than as incidental implementation noise.

ACM Reference Format:
Juan Manuel Rodriguez, Oleg Lesota, and Antonela Tommasel. 2026. Training seeds and model-selection stability in recommender-system evaluation. In *20th ACM Conference on Recommender Systems (RecSys '26), September 27-October 02, 2026, Minneapolis, MN, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3773078.3841289

1 Introduction

Recent reproducibility studies have questioned how confidently progress can be claimed in recommender system research. Prior work has shown that results may depend on baseline tuning, implementation details, data splitting, sampled evaluation, model initialization, and training-time sampling choices [1, 2, 7, 8, 11]. However, even under a fixed data partition, hyperparameters, and evaluation protocol, changing the training seed may affect both reported metrics and model-selection conclusions.

This paper focuses on *training-seed variation under an otherwise fixed experimental setup*. We fix the data partition and vary a single training seed across hyperparameter configurations. We interpret the seed as controlling the stochastic components of the full training run, rather than only parameter initialization. Our goal is not to attribute variability to a specific internal mechanism, but to examine whether repeated training-seed changes the conclusions researchers would draw from an experiment.

We examine whether training-seed variation affects user-level evaluation results, whether comparisons observed on validation data are preserved on test data, whether the ranking of hyperparameter configurations remains stable, and how much performance is lost when selecting a configuration based on validation results. This framing distinguishes detectable variation from practical instability. A seed may change user-level metric distributions without changing the selected configuration, while near-tied validation results may still affect model selection.

2 Methodology

Our methodology isolates training-seed variability under a fixed data partition and evaluates its effect on performance estimates, model selection decisions and recommendation lists[1].

Datasets. We use three benchmark datasets: Movielens-1M [3], Steam [6], and Amazon All Beauty 2023 [5]. Interactions are ordered temporally and split by user into train, validation and test sets using an 80%/10%/10% split.

Recommendation models. We evaluate four widely used recommender models, covering both matrix-factorization and sequential neural approaches: BPR [9], NeuMF [4], BERT4Rec [10], and SASRec [6]. BPR uses 3 embedding-size configuration. NeuMF uses 9 combinations of MF embedding size and MLP hidden layers. BERT4Rec and SASRec each use 12 configurations varying hidden layer size, number of layers, and attention heads[2].

Training procedure. All models are trained for up to 300 epochs for each configuration using early stopping with a patience of 10 epochs. For each model–configuration pair, we repeat training with the fixed seed set $\{1, \ldots, 10\}$ (chosen a priori and not based on performance to avoid seed tuning) and retain the checkpoint with the best validation performance. Unless otherwise specified, all remaining hyperparameters are kept at their RecBole defaults [12].

Evaluation focus. We analyze training-seed variation at four complementary levels, using nDCG@10 as the primary ranking metric[3]. Metric-level sensitivity. For each model and hyperparameter configuration, we compute user-level recommendation scores for each training seed. Since the same users are evaluated across seeds, we treat users as repeated observations and apply the Friedman test to compare the per-user metric distributions obtained under different

RecSys '26, Minneapolis, MN, USA

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/10.1145/3773078.3841289

[1] More details, code and results: https://github.com/knife982000/RecSeed
[2] Full hyperparameter spaces are provided in the companion repository.
[3] Recall@10, MRR@10, Hit@10, and Precision@10 can be found in the repository.

seeds. The analysis captures whether training-seed variation is detectable before considering its effects on model-selection decisions.
Model-selection stability. We evaluate whether seed variation affects the configuration selected from validation results. For each training seed, we identify the best validation configuration and compare it with the best observed under the corresponding evaluation setting. We report the performance gap (%) for validation–validation and validation–test comparisons, distinguishing instability within validation from that transferred to the final test evaluation.
Validation–test consistency. We evaluate whether pairwise configuration comparisons on validation are preserved on test. For each configuration pair, we estimate how often validation wins transfer to test. This indicates whether validation provides a reliable signal for ranking configurations, beyond their performance gap.
Recommendation-list agreement. We examine whether different seeds produce similar top-k recommendation lists. For each configuration, we compare lists generated by pairs of seeds using Jaccard similarity and Average Overlap@k: $AO@k = \frac{1}{k}\sum_{i=1}^{k}\frac{|R_1@i \cap R_2@i|}{i}$, where $R_x@i$ are the top-i recommendations by seed x. This captures whether seeds affect the recommendations actually produced.

3 Experimental results

Metric-level sensitivity. Figure 1 illustrates metric-level sensitivity for NeuMF, SASRec and BERT4Rec, using test nDCG@10 obtained by each seed for every hyperparameter configuration. The contrast shows that seed variation can take different forms. Across datasets, training-seed effects are often detectable at the user-score level, but their practical interpretation varies. On ML-1M, Friedman tests show clear metric-level sensitivity for BERT4Rec and SASRec, partial sensitivity for NeuMF, and limited sensitivity for BPR. Steam shows the strongest statistical signal, with Friedman tests rejecting the null for all configurations of all models on validation and test. However, statistical significance does not necessarily imply model-selection instability. For BERT4Rec and SASRec, seed effects often occur within clearly separated configuration regions, while NeuMF is the clearest flatter case, with stronger overlap across configurations. Overall, *metric-level sensitivity is a warning signal, but whether it affects experimental conclusions depends on the size of seed-induced variation relative to the gaps between configurations.*

Model-selection stability. Figure 2 shows three representative performance-gap cases: NeuMF on Amazon, where both validation–validation and validation–test gaps are high; NeuMF on Steam, where both gaps remain low; and BPR on Amazon, where validation–validation gaps are zero but validation–test gaps are non-zero. Overall, the performance-gap analysis shows that seed variation can affect model selection in different ways. In some cases, different seeds select different validation configurations, but the loss remains small because the competing configurations perform similarly (e.g., NeuMF on Steam). In other cases, different seeds lead to large gaps between the validation-selected configuration and the best observed configuration (e.g., NeuMF and BERT4Rec on Amazon). Finally, validation selection can be stable, but the selected configuration may not transfer as well to test (e.g., BPR on Amazon). Thus, *model-selection stability depends not only on whether the selected configuration changes across seeds, but also on the performance gap and on the alignment between validation and test behavior.*

Validation–test consistency. Results show that validation–test transfer is both dataset- and model-dependent. On ML-1M, validation comparisons are generally informative, although not uniformly so. BERT4Rec and SASRec show mostly high consistency, BPR shows moderate-to-high consistency, and NeuMF is the weakest case. Amazon shows a weaker and more heterogeneous transfer pattern, where several validation winners do not remain better on test. This helps explain the larger validation–test performance gaps observed. Instability comes not only from different seeds selecting different configurations, but also from validation rankings being less reliably preserved on test. BPR provides a useful contrast, as validation selection can be stable while the selected configuration still fails to transfer consistently to test. Steam shows the opposite tendency. Validation comparisons usually transfer well to test, especially for NeuMF, although SASRec remains more heterogeneous. Overall, validation-based model selection depends not only on the stability of the selected configuration across seeds, but also on whether validation comparisons preserve test ordering. Thus, *high seed sensitivity does not necessarily imply large selection losses when competing configurations perform similarly, but weak validation–test transfer can make validation-based choices unreliable even when validation selection appears stable.*

Recommendation-list agreement. Figure 3 illustrates this analysis for BPR on the three datasets, using Jaccard@k and AO@k agreement across pairs of seeds on the test split. The contrast is substantial, BPR produces moderately similar lists on ML-1M, almost disjoint lists on Amazon All Beauty, and highly consistent lists on Steam. Across datasets, AO@k is consistently higher than Jaccard@k, indicating that different seeds preserve more agreement near the top of the ranking than across the full top-k set. However, agreement remains below 1 in all cases, so repeated runs are not list-invariant. List-level stability is also strongly dataset-dependent. On ML-1M, agreement is moderate to high, with SASRec producing the most stable lists and NeuMF the least stable ones. On Amazon, stability is weakest, with BPR producing almost no overlap across seeds and BERT4Rec also showing low agreement, while SASRec is substantially more stable. On Steam, agreement is generally high across models, especially for BPR, SASRec, and NeuMF. Overall, *these results show that metric-level sensitivity, model-selection instability, and list-level instability are related but distinct effects of training seeds. Importantly, similar aggregate performance may still correspond to different recommendation lists, affecting user exposure.*

4 Conclusions

This work shows that training seeds can affect recommender evaluation at several levels. Across datasets and models, changing only the training seed can produce detectable differences in user-level scores, influence validation-based configuration choices, and change the top-k recommendation lists. Nonetheless, these effects do not always have the same practical meaning. Some are statistically detectable without substantially changing model-selection outcomes, while others reveal weaker validation–test transfer or different lists under similar aggregate performance. Overall, these results suggest that training seeds should be treated as part of the evaluation protocol rather than as incidental implementation noise. Reporting seed variation is especially important when configurations are close in performance, when validation and test behavior differ, or when conclusions rely on a single aggregate metric.

This study has several limitations. First, the training seed is treated as the seed of the full training run. Thus, the analysis captures run-to-run variability, but does not attribute this variability to a specific stochastic mechanism such as initialization, mini-batch ordering, dropout, latent sampling or training-time negative sampling. Second, data partition is fixed by design, which allows us to focus on training-seed variation. Future work should disentangle individual sources of training randomness, jointly analyze data-split, training and evaluation-sampling seeds, and extend the analysis to additional models, datasets, and beyond accuracy outcomes.

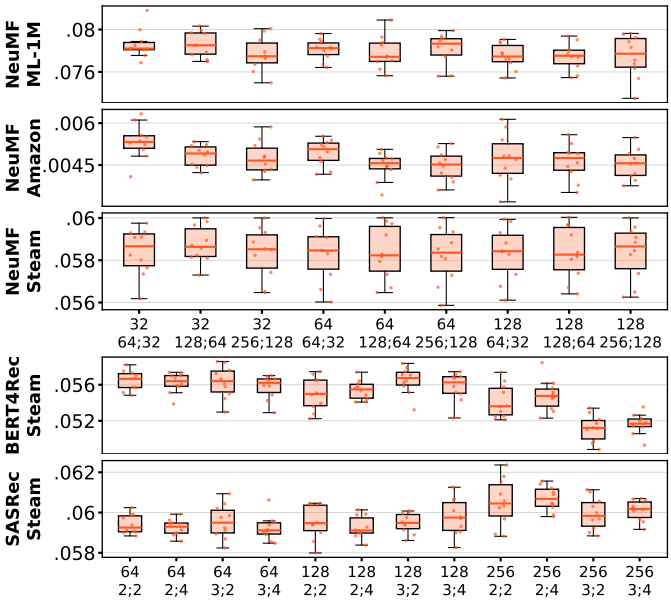

Figure 1: Per-seed test NDCG@10: NeuMF on the three datasets, then BERT4Rec and SASRec on Steam. Boxplots show mean NDCG@10 across training seeds. Labels on each group's bottom panel: MF embedding size over MLP layer sizes (NeuMF); hidden layers' size; number of heads (BERT4Rec/SASRec).

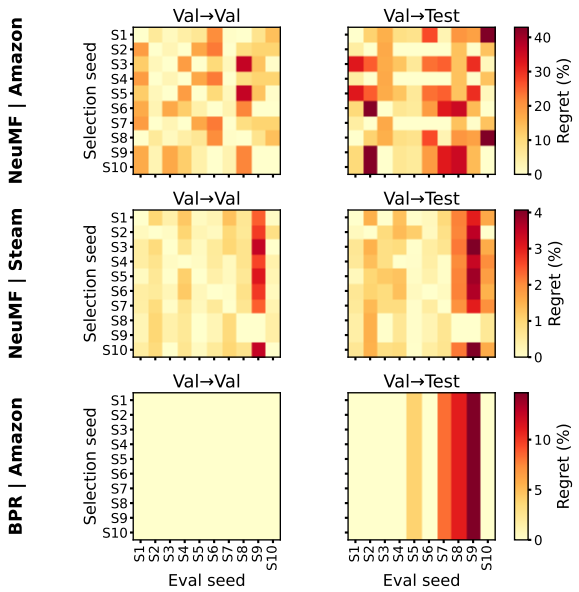

Figure 2: Performance gaps from validation-based configuration selection. Rows: seed used to select the best validation configuration; columns: evaluation seed. Colour encodes relative regret (%), with one shared scale per row.

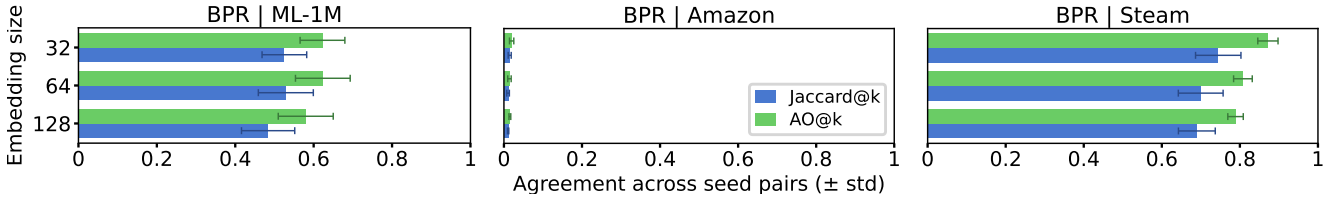

Figure 3: Cross-seed recommendation-list agreement for BPR on the test split.

References

[1] Edoardo D'Amico, Giovanni Gabbolini, Cesare Bernardis, and Paolo Cremonesi. 2022. Analyzing and improving stability of matrix factorization for recommender systems. *Journal of Intelligent Information Systems* 58, 2 (2022), 255–285.

[2] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. 2019. Are we really making much progress? A worrying analysis of recent neural recommendation approaches. In *Proceedings of the 13th ACM RecSys*. 101–109.

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

[4] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web* (Perth, Australia). 173–182.

[5] Yupeng Hou, Jiacheng Li, Xiangjun Fu, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. 2026. Bridging Language and Items for Retrieval and Recommendation: Benchmarking LLMs as Semantic Encoders. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (Eds.). 3251–3265.

[6] Wang-Cheng Kang and Julian McAuley. 2018. Self-Attentive Sequential Recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*. 197–206. doi:10.1109/ICDM.2018.00035

[7] 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*. 1748–1757.

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

[9] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback *(UAI '09)*. AUAI Press, Arlington, Virginia, USA, 452–461.

[10] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (Beijing, China). 1441–1450.

[11] Lukas Wegmeth, Tobias Vente, Lennart Purucker, and Joeran Beel. 2023. The Effect of Random Seeds for Data Splitting on Recommendation Accuracy.. In *Perspectives@ RecSys*.

[12] Lanling Xu, Zhen Tian, Gaowei Zhang, Junjie Zhang, Lei Wang, Bowen Zheng, Yifan Li, Jiakai Tang, Zeyu Zhang, Yupeng Hou, Xingyu Pan, Wayne Xin Zhao, Xu Chen, and Ji-Rong Wen. 2023. Towards a More User-Friendly and Easy-to-Use Benchmark Library for Recommender Systems. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2837–2847.