SL4SR: SELECTIVE LEARNING FOR SEQUENTIAL RECOMMENDATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in deep learning have substantially improved sequential recommendation (SR) by enabling more expressive architectures to capture complex user behavior patterns. However, the increased model capacity can also make SR models more prone to fitting incidental, transient, or weakly informative interactions that are less generalizable to future user preferences. This issue is further amplified by the prevailing training paradigm in SR, where standard objectives typically treat all training instances uniformly, regardless of their usefulness for improving the current model. To address this limitation, we propose **SL4SR**, a simple and effective **S**elective **L**earning framework for **S**equential **R**ecommendation. SL4SR leverages a pre-trained estimator as a reference model and compares its sample-wise loss with that of the current backbone to estimate the learning utility of each training instance. Based on this estimator-guided score, the model selectively updates on informative high-score instances while filtering out instances that are less likely to improve generalization. The proposed framework is model-agnostic and introduces no additional inference overhead. Extensive experiments on six real-world datasets with four representative SR backbones demonstrate the effectiveness of SL4SR, yielding average improvements of 16.97%, 17.65%, 17.07%, and 4.85% for SASRec, GRU4Rec, FMLPRec, and Mamba4Rec, respectively.

1 INTRODUCTION

Sequential recommendation (SR) aims to predict the next item a user is likely to interact with based on the user's historical interaction sequence (Hidasi et al., 2016; Kang & McAuley, 2018; Tang & Wang, 2018; Chang et al., 2021; Xie et al., 2022; Zhou et al., 2022; Rajput et al., 2023; Li et al., 2023; Xu et al., 2024; Heo et al., 2026). As a core component of modern recommender systems, SR has become increasingly important in a wide range of online services, including e-commerce, video streaming, and content platforms, where accurate next-item prediction directly affects user engagement, conversion, and long-term platform growth (Mahajan et al., 2023; Ko et al., 2022; Yin et al., 2012). Because user preference is inherently dynamic and sequential, SR has attracted sustained attention as a fundamental problem in recommender systems.

With the rise of deep learning, early neural approaches employed recurrent neural networks (Hidasi et al., 2016; Li et al., 2017) and convolutional neural networks (Tang & Wang, 2018) to capture local or temporal dependencies in user behavior sequences. Subsequently, Transformer-based models (Kang & McAuley, 2018; Sun et al., 2019; de Souza Pereira Moreira et al., 2021) emerged as a dominant paradigm owing to their ability to capture long-range dependencies and global contextual patterns through the self-attention mechanism (Vaswani et al., 2017). As a result, modern SR models are capable of leveraging much richer sequential signals from user-item interaction histories than before. This increasing modeling capacity has substantially improved recommendation quality and enabled more effective sequential modeling.

However, stronger modeling capacity also introduces an important challenge. In practice, user interaction sequences do not consist solely of clean and representative preference signals (Agichtein et al., 2006; Said et al., 2012; Zhang et al., 2020; Zhou et al., 2022). Real-world sequences often contain interactions that are incidental, transient, exploratory, or only weakly related to the user's stable preference. As illustrated in Figure 1, a user who consistently watches action movies may

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

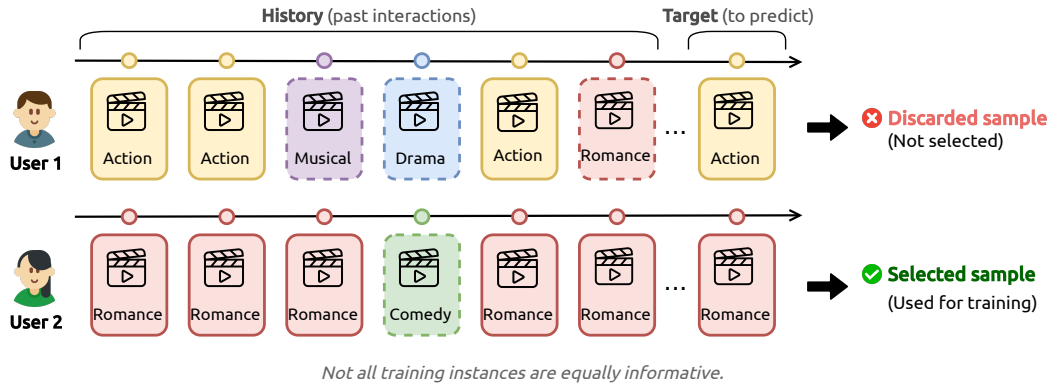


Figure 1: Illustration of sample-wise informativeness in sequential recommendation. User interaction sequences may contain incidental or less representative behaviors, making some training instances less useful for optimization than others.

occasionally watch a romance movie due to temporary curiosity, social influence, or contextual factors unrelated to long-term preference. Such interactions are not necessarily erroneous observations, but they may provide weak or misleading supervision for predicting future behavior. As SR models become more expressive, they fit not only stable preference patterns but also these less generalizable interaction signals.

This challenge is further amplified by the prevailing training paradigm in SR. While modern SR models can differentiate the relative importance of interactions within a sequence through mechanisms such as self-attention, the optimization objective still typically treats training instances equally (Rendle et al., 2012; Hidasi et al., 2016; Kang & McAuley, 2018; Liu et al., 2024a). As a result, an instance derived from a highly representative behavioral context and another derived from a weakly informative or less stable context often contribute equally to the training objective. Although this strategy is simple and broadly effective, it overlooks a critical question: are all training instances equally worth learning at a given stage of training? We argue that the answer is often no. Some instances may contain highly learnable and generalizable patterns that the current model has not yet captured, whereas others may offer limited benefit for improving generalization.

This observation motivates us to revisit SR training from the perspective of sample-wise learnability. Rather than uniformly optimizing all training instances, we ask whether the optimization process can be guided toward instances that are more useful for improving the current model. Our key intuition is that the value of a training instance should be assessed not only by its current loss, but also by the extent to which that loss remains reducible relative to a reference model. A large loss alone does not necessarily imply that an instance deserves greater emphasis, as it may reflect limited generalizability rather than insufficient learning. In contrast, an instance whose loss remains substantially higher than that of a reference model may still contain learnable structure that the backbone has not yet captured.

To this end, we propose a selective learning framework for sequential recommendation. The key idea is to leverage a pre-trained estimator as a reference model and compare its sample-wise loss with that of the current backbone. This comparison provides an estimator-guided proxy for sample-wise learnability. Intuitively, if the current backbone performs substantially worse than the estimator on a particular training instance, that instance may still contain reducible loss and should therefore be prioritized during optimization. In contrast, if the loss gap between the backbone and the estimator is relatively small, the corresponding instance may offer limited additional learning benefit for the current model. Based on this intuition, our framework selectively prioritizes training instances with higher estimator-relative learnability while filtering out those that appear less useful for improving generalization.

Our main contributions are summarized as follows:

- We revisit sequential recommendation from the perspective of sample-wise learnability and highlight the limitation of uniformly optimizing all training instances, despite the increasing modeling capacity of modern SR architectures.

- We propose a selective learning framework for sequential recommendation that leverages a pre-trained estimator as a reference model and uses estimator-guided loss comparison as a proxy for sample-wise learnability, enabling the model to prioritize more learnable training instances during optimization.
- We show that the proposed framework can be seamlessly integrated into representative sequential recommendation models and consistently improves recommendation performance across datasets, demonstrating the effectiveness of estimator-guided selective learning in SR.

2 Related Work

2.1 Sequential Recommender Systems

Sequential recommendation aims to predict the next item a user is likely to interact with by modeling sequential dependencies in historical behavior. Early approaches mainly focused on transition-based modeling, as represented by factorized personalized Markov chains (FPMC) (Rendle et al., 2010) and its higher-order extension Fossil (He & McAuley, 2016). With the development of deep learning, neural sequence encoders became widely adopted for SR. GRU4Rec (Hidasi et al., 2016) uses gated recurrent units to capture temporal dynamics in user behavior sequences, while Caser (Tang & Wang, 2018) applies convolutional filters over item-embedding sequences to extract local sequential patterns. Attention-based architectures have subsequently become a major line of research. SASRec (Kang & McAuley, 2018) introduces causal self-attention to model long-range dependencies over the entire interaction history. BERT4Rec (Sun et al., 2019), inspired by masked language modeling, adopts bidirectional self-attention with a masked-item prediction objective. S^3-Rec (Zhou et al., 2020) further improves representation learning through self-supervised objectives defined at multiple granularities. Beyond attention mechanisms, recent studies have explored alternative sequence modeling paradigms. FMLPRec (Zhou et al., 2022) replaces self-attention with learnable frequency-domain filters, and BSARec (Shin et al., 2024) combines Fourier transforms with attention to enhance frequency-aware sequential representations. More recently, Mamba4Rec (Liu et al., 2024a) leverages selective state space models to capture long-range user preferences efficiently without relying on conventional self-attention.

2.2 Learning Strategy for Sequential Recommendation

Beyond architectural advances, prior studies have also attempted to improve sequential recommendation through better training objectives and optimization strategies. Many existing methods rely on point-wise or pair-wise objectives for next-item prediction, such as cross-entropy and Bayesian Personalized Ranking (BPR) (Rendle et al., 2012; Hidasi et al., 2016; Kang & McAuley, 2018). More recently, auxiliary objectives such as self-supervised learning and masked-item prediction have been introduced to improve representation quality and robustness under sparse supervision (Xie et al., 2022; Zhou et al., 2020; Sun et al., 2019). Another line of work explores optimization-oriented improvements, such as negative sampling and hard negative mining (Rendle et al., 2012; Zhang et al., 2013). Although these approaches improve optimization in different ways, they generally still assume that all training instances should contribute to learning. As a result, the question of whether different training instances should be treated differently during optimization remains largely underexplored in sequential recommendation. In the broader machine learning literature, related ideas have been studied under curriculum learning, self-paced learning, hard example mining, data selection, and time-series selective learning (Bengio et al., 2009; Kumar et al., 2010; Schroff et al., 2015; Mindermann et al., 2022; Fu et al., 2025). These methods share the intuition that not all training examples are equally beneficial throughout training. Inspired by this broader perspective, we introduce a selective learning framework tailored to sequential recommendation, which explicitly identifies and prioritizes training instances that are estimated to be more useful for improving the current model. Unlike conventional training strategies that rely on predefined difficulty measures, model confidence, or generic sample-selection criteria, SL4SR introduces a frozen estimator trained on the full training set as a reference for assessing each instance relative to the current backbone. The resulting loss gap provides a model-relative signal of how differently an instance is fitted by the current backbone and the reference estimator. SL4SR then uses this signal to selectively determine which instances contribute to each optimization step, establishing estimator-guided sample selection as a distinct learning strategy for sequential recommendation.

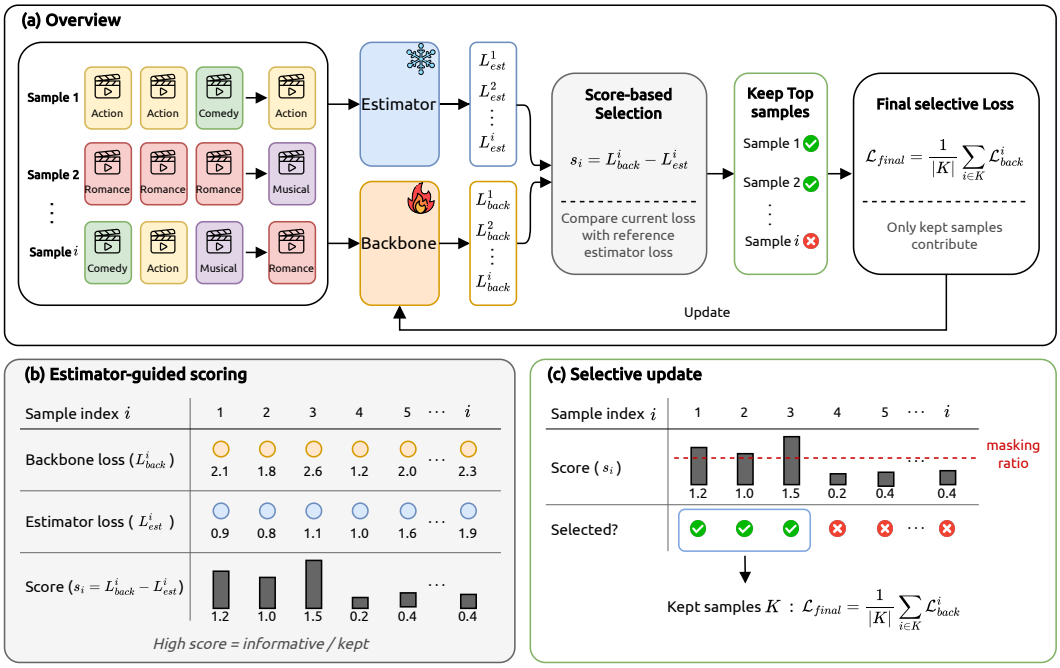


Figure 2: Illustration of selective learning framework.

3 Selective Learning

3.1 Overview

We propose a model-agnostic selective learning framework for sequential recommendation that operates in a sample-wise manner. The core idea is to avoid having all training instances participate equally in every optimization step and instead prioritize those that appear more learnable for the current backbone model. As illustrated in Figure 2, our framework selectively filters out training instances that are less likely to provide useful optimization signals, while retaining those with greater potential to improve the model. To achieve this, we introduce a pre-trained estimator as a frozen reference model and compare its sample-wise loss with that of the trainable backbone on the same mini-batch. This comparison provides an estimator-guided proxy for sample-wise learnability. Intuitively, if the backbone incurs a substantially larger loss than the estimator on a particular training instance, the resulting loss gap indicates that the current backbone fits that instance considerably worse than a fully trained reference model. Such a discrepancy suggests that the instance may still provide useful learning signals for the current backbone and is therefore assigned a higher priority during optimization. In contrast, if the loss gap is relatively small, the backbone already exhibits a level of fit closer to the estimator, and the instance is assigned a lower priority in the current update.

3.2 Estimator-Guided Scoring

Let $\{x_1, x_2, \ldots, x_N\}$ denote a set of training samples, where each sample corresponds to an interaction sequence and its associated supervision target. For each sample x_i, we evaluate two models: a trainable backbone model $f(\cdot;\theta)$ and a frozen estimator $g(\cdot;\phi)$. In our default setting, the estimator is a pre-trained model of the same architecture as the backbone, trained beforehand on the same dataset and fixed during selective learning.

For each sample x_i, we compute the sample-wise losses of the backbone and the estimator, denoted by $\mathcal{L}^i_{back}$ and $\mathcal{L}^i_{est}$, respectively. The backbone loss reflects how well the current model explains the given training instance, whereas the estimator loss provides a reference signal from the pre-trained model. Based on these two losses, we define an estimator-guided score for each sample as

$$s_i = \mathcal{L}^i_{back} - \mathcal{L}^i_{est}, \tag{1}$$

where s_i denotes the score of the i-th sample.

The score in Eq. 1 is designed to quantify the fit discrepancy between the current backbone and the fixed estimator. If the backbone incurs a substantially larger loss than the estimator on sample x_i, the resulting score becomes large, indicating that the current backbone fits the instance less well than the reference model. Conversely, a small loss gap indicates that the backbone and estimator exhibit more similar fit on the instance. We use this estimator-relative discrepancy as a practical criterion for ranking samples within each mini-batch and prioritize high-score instances for the current update.

3.3 Selective Sample Update

Once the estimator-guided scores $\{s_1, s_2, \ldots, s_N\}$ are obtained, we select only a subset of informative samples for optimization. Specifically, let ρ denote the masking ratio, which determines the proportion of low-score samples to exclude from the current update. Based on the scores, we retain the top-scoring $(1-\rho)N$ samples and denote the resulting kept set by

$$K = \mathrm{TopK}\big(\{s_i\}_{i=1}^{N}, (1-\rho)N\big), \tag{2}$$

where K contains the indices of the retained high-score samples.

Given the kept set K, we define the final selective loss using only the backbone losses of the retained samples:

$$\mathcal{L}_{\text{final}} = \frac{1}{|K|} \sum_{i \in K} \mathcal{L}_{back}^{i}, \tag{3}$$

where $|K|$ denotes the number of retained samples. In this way, only high-score samples contribute to the optimization objective, while low-score samples are excluded from the current update.

3.4 Training and Inference

Problem Definition. Let $\mathcal{S}^u = [v_1^u, v_2^u, \ldots, v_{|\mathcal{S}^u|}^u]$ denote the item interaction sequence of user u, where $v_t^u \in \mathcal{V}$ is the item interacted with at step t, and $\mathcal{V}$ is the item set. Sequential recommendation aims to predict the next item conditioned on the user's historical interaction sequence. Formally, given the interaction sequence $\mathcal{S}^u$, the model estimates the conditional probability of each candidate item $v \in \mathcal{V}$ as

$$\hat{y}_v = p(v_{|\mathcal{S}^u|+1}^u = v \mid \mathcal{S}^u), \tag{4}$$

where $\hat{y}_v$ denotes the predicted score or probability of item v being the next interaction.

Training. During training, we first obtain a pre-trained estimator on the same training dataset and then freeze its parameters. The trainable backbone and the frozen estimator are both evaluated on the same training samples to obtain the sample-wise losses $\mathcal{L}_{back}^{i}$ and $\mathcal{L}_{est}^{i}$, respectively. Based on the estimator-guided scoring and selective update, only the retained high-score samples contribute to the final training objective, and gradients are backpropagated to the backbone alone. The detailed training procedure is summarized in Appendix A.1.

Inference. During inference, the estimator is discarded and only the trained backbone is used for next-item prediction. Therefore, the proposed selective learning framework introduces no additional inference overhead.

Table 1: Statistics of the datasets after preprocessing.

Datasets	# Users	# Items	# Interactions	Avg. length	Sparsity
Beauty	22,363	12,101	198,502	8.9	99.93%
Sports	35,598	18,357	296,337	8.3	99.93%
Instruments	1,429	900	10,261	7.2	99.20%
LastFM	1,090	3,646	52,551	48.2	98.68%
ML-1M	6,041	3,417	999,611	165.5	95.16%
Yelp	30,431	20,033	316,354	10.4	99.95%

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Dataset. We conduct experiments on six widely used real-world benchmark datasets for sequential recommendation: Amazon Beauty (Beauty), Amazon Sports and Outdoors (Sports), Amazon Musical Instruments (Instruments), LastFM, MovieLens-1M (ML-1M), and Yelp. Following prior work (Kang & McAuley, 2018; Zhou et al., 2020), we sort user–item interactions in chronological order to construct user interaction sequences, which are then split into training, validation, and test sets. Table 1 reports the statistics of the resulting datasets. More detailed descriptions of the datasets are provided in Appendix A.2.

Baselines. We evaluate the proposed framework on four representative sequential recommendation backbones: SASRec (Kang & McAuley, 2018), GRU4Rec (Hidasi et al., 2016), FMLPRec (Zhou et al., 2022), and Mamba4Rec (Liu et al., 2024a). SASRec is a widely used self-attention based Transformer model, GRU4Rec represents recurrent neural network based sequential recommendation, FMLPRec captures sequential patterns through frequency-domain filtering, and Mamba4Rec adopts a state space modeling paradigm. These backbones span diverse architectural families in the SR literature, making them suitable testbeds for evaluating the model-agnostic nature of the proposed framework. Further implementation details of the baselines are provided in Appendix A.3.

Evaluation Metrics. For evaluation, we follow the leave-one-out protocol adopted in recent studies (Rajput et al., 2023; Kang & McAuley, 2018). We report Hit Ratio at k (HR@k) and Normalized Discounted Cumulative Gain at k (NDCG@k) with $k \in \{5, 10, 20\}$. To ensure a fair and comprehensive comparison, all methods are evaluated by ranking over the full item set without negative sampling (Krichene & Rendle, 2020).

Implementation Details. For the proposed selective learning framework, we tune the masking ratio ρ over $\{0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. Across all models, we set the maximum sequence length to 50, the hidden embedding size to 64, and the batch size to 256. We train all models using the Adam optimizer (Kingma, 2014) with a learning rate of 0.001 and apply early stopping based on validation performance. For backbone-specific settings, SASRec uses 2 Transformer layers and 2 attention heads, while Mamba4Rec uses an SSM state expansion factor of 32, a 1D convolution kernel size of 4, and a block expansion factor of 2 for linear projections. Additional implementation details are provided in the code [1].

4.2 OVERALL PERFORMANCE

Table 2 reports the performance comparison without and with selective learning. Overall, applying selective learning consistently improves performance across all datasets and evaluation metrics. When averaged over all dataset–metric combinations, selective learning yields improvements of 16.97% for SASRec, 17.65% for GRU4Rec, 17.07% for FMLPRec, and 4.85% for Mamba4Rec. Among the four backbones, GRU4Rec shows the largest average improvement, while SASRec and FMLPRec also achieve substantial and consistent gains. These results indicate that the effectiveness of selective learning is not limited to a specific model family. Instead, it generalizes well across diverse sequential recommendation architectures, including self-attention based Transformers, recurrent neural networks, frequency-based filtering models, and state space models. This supports the model-agnostic nature of the proposed framework and demonstrates that estimator-guided sample selection provides broadly useful training signals across different sequential modeling paradigms.

4.3 ABLATION STUDY

We conduct an ablation study to verify the effectiveness of the proposed selective learning strategy. As shown in Table 3, we compare our full method with three variants: (1) Random Mask, which randomly masks training instances under the same masking ratio, (2) Top-1, which updates the model using only the highest-scored instance in each batch, and (3) RHO-LOSS (Mindermann et al., 2022), which prioritizes samples based on the difference between the current model loss and a reference-model loss estimated from a held-out subset. Random masking improves over the original backbone in some cases, suggesting that uniformly optimizing all training instances can sometimes

[1] https://anonymous.4open.science/r/SL4SR-B604

Table 2: Performance comparison without/with selective learning (SL). Better results are in **bold**, 'Improv.' indicates the improvements caused by selective learning.

Datasets	Metric	SASRec	+SL	Improv.	GRU4Rec	+SL	Improv.	FMLPRec	+SL	Improv.	Mamba4Rec	+SL	Improv.
Beauty	HR@5	0.0298	**0.0368**	23.49%	0.0158	**0.0198**	25.32%	0.0320	**0.0375**	17.19%	0.0684	**0.0705**	3.07%
	HR@10	0.0487	**0.0567**	16.43%	0.0284	**0.0334**	17.61%	0.0537	**0.0592**	10.24%	0.0962	**0.0974**	1.25%
	HR@20	0.0755	**0.0851**	12.72%	0.0482	**0.0543**	12.66%	0.0853	**0.0901**	5.63%	0.1313	**0.1352**	2.97%
	NDCG@5	0.0189	**0.0237**	25.40%	0.0104	**0.0126**	21.15%	0.0199	**0.0241**	21.11%	0.0488	**0.0500**	2.46%
	NDCG@10	0.0250	**0.0301**	20.40%	0.0144	**0.0169**	17.36%	0.0269	**0.0311**	15.61%	0.0577	**0.0587**	1.73%
	NDCG@20	0.0317	**0.0373**	17.67%	0.0194	**0.0222**	14.43%	0.0348	**0.0389**	11.78%	0.0666	**0.0682**	2.40%
Sports	HR@5	0.0173	**0.0199**	15.03%	0.0116	**0.0141**	21.55%	0.0168	**0.0200**	19.05%	0.0397	**0.0410**	3.27%
	HR@10	0.0280	**0.0308**	10.00%	0.0204	**0.0221**	8.33%	0.0291	**0.0311**	6.87%	0.0569	**0.0580**	1.93%
	HR@20	0.0436	**0.0485**	11.24%	0.0315	**0.0365**	15.87%	0.0465	**0.0486**	4.52%	0.0799	**0.0818**	2.38%
	NDCG@5	0.0111	**0.0133**	19.82%	0.0073	**0.0096**	31.51%	0.0107	**0.0125**	16.82%	0.0280	**0.0291**	3.93%
	NDCG@10	0.0146	**0.0168**	15.07%	0.0101	**0.0122**	20.79%	0.0146	**0.0161**	10.27%	0.0335	**0.0345**	2.99%
	NDCG@20	0.0185	**0.0212**	14.59%	0.0129	**0.0158**	22.48%	0.0190	**0.0205**	7.89%	0.0393	**0.0405**	3.05%
Instruments	HR@5	0.0742	**0.0826**	11.32%	0.0483	**0.0490**	1.45%	0.0616	**0.0798**	29.55%	0.0798	**0.0812**	1.75%
	HR@10	0.1106	**0.1155**	4.43%	0.0742	**0.0756**	1.89%	0.1008	**0.1127**	11.81%	0.1169	**0.1225**	4.79%
	HR@20	0.1561	**0.1596**	2.24%	0.1169	**0.1211**	3.59%	0.1435	**0.1554**	8.29%	0.1547	**0.1638**	5.88%
	NDCG@5	0.0491	**0.0545**	11.00%	0.0333	**0.0347**	4.20%	0.0393	**0.0495**	25.95%	0.0566	**0.0570**	0.71%
	NDCG@10	0.0605	**0.0650**	7.44%	0.0414	**0.0429**	3.62%	0.0520	**0.0600**	15.38%	0.0688	**0.0703**	2.18%
	NDCG@20	0.0721	**0.0760**	5.41%	0.0519	**0.0542**	4.43%	0.0629	**0.0707**	12.40%	0.0783	**0.0804**	2.68%
LastFM	HR@5	0.0339	**0.0459**	35.40%	0.0257	**0.0266**	3.50%	0.0339	**0.0495**	46.02%	0.0431	**0.0495**	14.85%
	HR@10	0.0532	**0.0670**	25.94%	0.0358	**0.0468**	30.73%	0.0514	**0.0706**	37.35%	0.0596	**0.0697**	16.95%
	HR@20	0.0789	**0.0927**	17.49%	0.0486	**0.0679**	39.71%	0.0798	**0.0991**	24.19%	0.0954	**0.0991**	3.88%
	NDCG@5	0.0212	**0.0311**	46.70%	0.0163	**0.0198**	21.47%	0.0231	**0.0346**	49.78%	0.0288	**0.0320**	11.11%
	NDCG@10	0.0272	**0.0380**	39.71%	0.0197	**0.0262**	32.99%	0.0285	**0.0413**	44.91%	0.0340	**0.0385**	13.24%
	NDCG@20	0.0337	**0.0443**	31.45%	0.0229	**0.0314**	37.12%	0.0355	**0.0486**	36.90%	0.0429	**0.0458**	6.76%
ML-1M	HR@5	0.1257	**0.1366**	8.67%	0.1068	**0.1207**	13.01%	0.1318	**0.1348**	2.28%	0.2043	**0.2108**	3.18%
	HR@10	0.1998	**0.2126**	6.41%	0.1753	**0.1959**	11.75%	0.2108	**0.2162**	2.56%	0.2950	**0.3043**	3.15%
	HR@20	0.3040	**0.3207**	5.49%	0.2765	**0.2992**	8.21%	0.3212	**0.3232**	0.62%	0.4099	**0.4179**	1.95%
	NDCG@5	0.0799	**0.0879**	10.01%	0.0685	**0.0783**	14.31%	0.0860	**0.0875**	1.74%	0.1375	**0.1420**	3.27%
	NDCG@10	0.1038	**0.1123**	8.19%	0.0907	**0.1026**	13.12%	0.1113	**0.1137**	2.16%	0.1667	**0.1721**	3.24%
	NDCG@20	0.1301	**0.1395**	7.23%	0.1162	**0.1287**	10.76%	0.1391	**0.1405**	1.01%	0.1958	**0.2007**	2.50%
Yelp	HR@5	0.0136	**0.0166**	22.06%	0.0121	**0.0157**	29.75%	0.0154	**0.0184**	19.48%	0.0228	**0.0244**	7.02%
	HR@10	0.0241	**0.0286**	18.67%	0.0207	**0.0264**	27.54%	0.0256	**0.0310**	21.09%	0.0394	**0.0420**	6.60%
	HR@20	0.0412	**0.0474**	15.05%	0.0369	**0.0431**	16.80%	0.0433	**0.0500**	15.47%	0.0654	**0.0697**	6.57%
	NDCG@5	0.0081	**0.0103**	27.16%	0.0074	**0.0095**	28.38%	0.0097	**0.0117**	20.62%	0.0145	**0.0155**	6.90%
	NDCG@10	0.0115	**0.0141**	22.61%	0.0101	**0.0129**	27.72%	0.0130	**0.0157**	20.77%	0.0198	**0.0212**	7.07%
	NDCG@20	0.0158	**0.0188**	18.99%	0.0142	**0.0171**	20.42%	0.0175	**0.0205**	17.14%	0.0264	**0.0282**	6.82%

hurt generalization. However, it does not consistently yield improvements because the removed instances are selected without considering their relative utility to the current model. The Top-1 variant shows substantially worse performance in most cases, indicating that overly aggressive selection leaves insufficient training signals for stable optimization. Compared with RHO-LOSS, our selective learning framework consistently achieves better performance across both backbones and all evaluated datasets and metrics. While RHO-LOSS uses a held-out reference model to estimate the irreducible component of the loss, our framework employs a fixed estimator trained on the full training set and uses the resulting loss gap as a model-relative signal for instance prioritization. The consistent gains over RHO-LOSS further support the effectiveness of the proposed estimator-guided formulation for selective optimization in sequential recommendation. Overall, the proposed selective learning strategy consistently achieves the best performance, supporting that its effectiveness stems from estimator-guided instance selection rather than from sample reduction alone.

Table 3: Ablation results for selective learning with SASRec and FMLPRec as backbones.

Methods	Beauty		Sports		Instruments		LastFM	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
SASRec	0.0487	0.0250	0.0280	0.0146	0.1106	0.0605	0.0532	0.0272
(1) Random Mask	0.0555	0.0296	0.0247	0.0128	0.0763	0.0320	0.0495	0.0259
(2) Top-1	0.0109	0.0051	0.0087	0.0045	0.0644	0.0395	0.0257	0.0173
(3) RHO-LOSS	0.0492	0.0251	0.0287	0.0151	0.0784	0.0434	0.0624	0.0367
(4) **With SL**	**0.0567**	**0.0301**	**0.0308**	**0.0168**	**0.1155**	**0.0650**	**0.0670**	**0.0380**
FMLPRec	0.0537	0.0269	0.0291	0.0146	0.1008	0.0520	0.0514	0.0285
(1) Random Mask	0.0545	0.0286	0.0264	0.0140	0.1099	0.0588	0.0459	0.0284
(2) Top-1	0.0097	0.0051	0.0070	0.0045	0.0812	0.0407	0.0073	0.0026
(3) RHO-LOSS	0.0508	0.0263	0.0293	0.0152	0.1043	0.0529	0.0587	0.0342
(4) **With SL**	**0.0592**	**0.0311**	**0.0311**	**0.0161**	**0.1127**	**0.0600**	**0.0706**	**0.0413**

4.4 Loss Analysis

We further analyze the training loss curves of SASRec to examine how selective learning affects the optimization process. In this analysis, the EST curve denotes the training loss of the pre-trained estimator, i.e., the standard model trained over all training instances, while the SL curve denotes the final selective loss computed only over the retained high-score instances. As shown in Figure 3, both EST and SL exhibit stable decreasing trends on Sports, LastFM, and Yelp, indicating that selective learning maintains stable optimization. SL maintains higher loss values than EST in most epochs. This is expected because the SL curve is computed on retained high-score instances, which are selected based on relatively large estimator-guided loss gaps, rather than averaged over all training instances. Despite the higher selective loss, SL achieves better recommendation performance in Table 2. This is particularly evident on LastFM, where EST rapidly reduces the training loss but yields lower recommendation accuracy than SL. These results suggest that aggressively minimizing the average training loss is not always aligned with better generalization in SR, because the standard objective reduces loss over all instances, including those derived from weakly informative or less generalizable behavioral contexts. In contrast, selective learning changes which training instances contribute to the update, allowing the model to focus more strongly on informative high-score instances that provide greater learning benefit to the current backbone. Additional loss curves for other backbone models are provided in Appendix B.1.

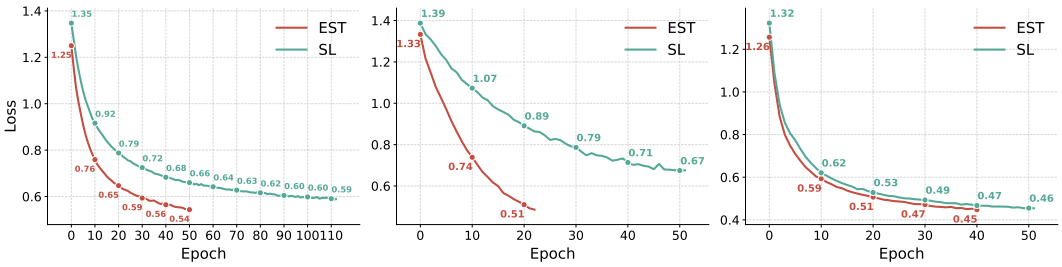


Figure 3: Training loss curves of SASRec under standard and selective learning.

4.5 Impact of Selective Mask Ratio

We investigate the impact of the selective mask ratio ρ, which controls the proportion of low-score samples excluded from the current update. A larger ρ indicates a more aggressive selection strategy, where fewer training samples are retained for optimization. Therefore, this hyperparameter directly determines how strongly the model focuses on high-score samples identified by the estimator-guided scoring criterion. As shown in Figure 4, the optimal selective mask ratio varies across different backbone models. This indicates that the desirable degree of sample selection depends on the learning dynamics and inductive bias of each architecture. Although the performance does not change monotonically with ρ, the most aggressive masking setting, $\rho = 0.9$, often yields lower performance than intermediate masking ratios. This suggests that excessively excluding training instances can reduce effective supervision and hinder stable optimization. These results highlight the importance of selecting an appropriate mask ratio and confirm that selective learning is most effective when it balances filtering less useful samples with preserving sufficient training signals. Further results are reported in Appendix B.2.

4.6 Cross-Estimator Analysis

To examine the flexibility of the proposed framework, we conduct a cross-estimator analysis where the estimator and the backbone use different architectures. Since different SR backbones may adopt different training ob-

Table 4: Cross-estimator analysis on the Beauty dataset. SASRec is used as the estimator for different backbone models.

Estimator	Metric	GRU4Rec	+SL	FMLPRec	+SL	Mamba4Rec	+SL
SASRec	HR@10	0.0284	**0.0314**	0.0537	**0.0573**	**0.0962**	0.0943
	NDCG@10	0.0144	**0.0160**	0.0269	**0.0302**	**0.0577**	0.0560

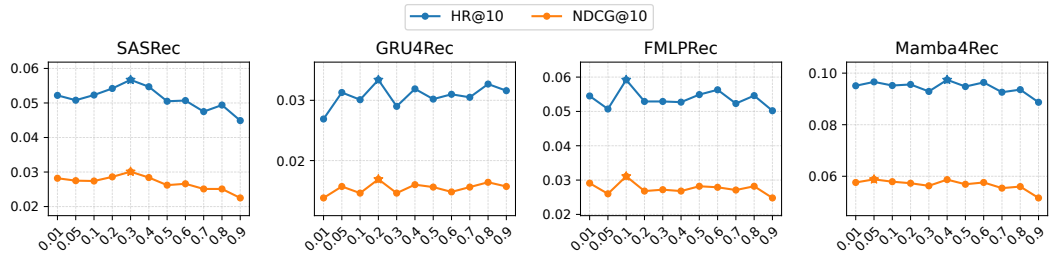


Figure 4: Impact of the selective mask ratio on the Beauty dataset across different backbone models.

jectives, their original sample-wise losses are not directly comparable. Therefore, we compute the selection score using a unified sample-wise BCE scoring loss across all estimator-backbone combinations. As shown in Table 4, using SASRec as the estimator improves both GRU4Rec and FMLPRec on HR@10 and NDCG@10, indicating that selective learning is not restricted to the same-architecture setting. In contrast, Mamba4Rec shows a slight performance drop, possibly due to the architectural and optimization mismatch between SASRec and Mamba4Rec, which can make the estimator-guided score less aligned with the backbone's learning dynamics. Overall, these results suggest that selective learning is most effective with a well-aligned estimator, while still showing potential in cross-estimator scenarios.

4.7 Long-tailed Recommendation

Long-tail distributions are prevalent in real-world recommender systems, where many users have limited interaction histories. To evaluate the effectiveness of selective learning under such long-tailed scenarios, we follow the protocol of (Kim et al., 2023; Liu et al., 2024b) and partition users based on interaction sequence length. Specifically, users are sorted by their sequence lengths, and the top 20% are defined as the head group, while the remaining users constitute the tail group, following the Pareto principle (Box & Meyer, 1986). Figure 5 presents the results on Beauty and Sports. Incorporating selective learning improves recommendation performance across backbone models and user groups, with gains also observed for tail users where interaction data is limited. These results show that the proposed training strategy remains effective under long-tailed user distributions. Moreover, SL consistently improves performance for head users, indicating that its benefits are not limited to sparse users but extend across both head and tail groups. Additional analysis is provided in Appendix B.4.

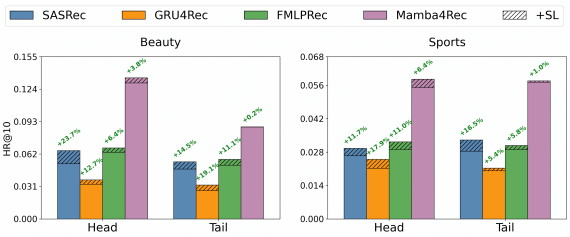


Figure 5: Performance comparison across user head and tail groups on Beauty and Sports datasets.

5 Conclusion

In this work, we introduce selective learning, a model-agnostic training strategy for sequential recommendation. Our framework leverages a pre-trained estimator as a reference model and compares its sample-wise loss with that of the current backbone to identify more informative and generalizable training instances. By optimizing the backbone only with selected high-score samples, selective learning guides the model toward instances with greater estimator-relative learnability. Extensive experiments on six real-world datasets and diverse backbone architectures demonstrate the effectiveness and broad applicability of the proposed framework.

AI USE STATEMENT

In this work, we used generative AI tools only to edit the manuscript for language, grammar, and readability. We did not use generative AI tools for research methodology, experiment design, implementation, data analysis, result interpretation, hypothesis formulation, or theoretical development. All AI-assisted edits were reviewed and verified by the authors. We take full responsibility for the final content of this work.

REFERENCES

Eugene Agichtein, Eric Brill, and Susan Dumais. Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 19–26, 2006.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.

George EP Box and R Daniel Meyer. An analysis for unreplicated fractional factorials. *Technometrics*, 28(1):11–18, 1986.

Jianxin Chang, Chen Gao, Yu Zheng, Yiqun Hui, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. Sequential recommendation with graph neural networks. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pp. 378–387, 2021.

Gabriel de Souza Pereira Moreira, Sara Rabhi, Jeong Min Lee, Ronay Ak, and Even Oldridge. Transformers4rec: Bridging the gap between nlp and sequential/session-based recommendation. In *Proceedings of the 15th ACM conference on recommender systems*, pp. 143–153, 2021.

Yisong Fu, Zezhi Shao, Chengqing Yu, Yujie Li, Zhulin An, Qi Wang, Yongjun Xu, and Fei Wang. Selective learning for deep time series forecasting. *arXiv preprint arXiv:2510.25207*, 2025.

Ruining He and Julian McAuley. Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th international conference on data mining (ICDM)*, pp. 191–200. IEEE, 2016.

Byungmoon Heo, Namjun Lee, Seonah Kim, and Jaekwang Kim. Fcrllm: Aligning llm with collaborative filtering for long-tailed sequential recommendation. In *Proceedings of the ACM Web Conference 2026*, pp. 6574–6585, 2026.

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. In *The International Conference on Learning Representations*, 2016.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pp. 197–206. IEEE, 2018.

Kibum Kim, Dongmin Hyun, Sukwon Yun, and Chanyoung Park. Melt: Mutual enhancement of long-tailed user and item for sequential recommendation. In *Proceedings of the 46th international ACM SIGIR conference on Research and development in information retrieval*, pp. 68–77, 2023.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Hyeyoung Ko, Suyeon Lee, Yoonseo Park, and Anna Choi. A survey of recommendation systems: recommendation models, techniques, and application fields. *Electronics*, 11(1):141, 2022.

Walid Krichene and Steffen Rendle. On sampled metrics for item recommendation. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1748–1757, 2020.

M Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. *Advances in neural information processing systems*, 23, 2010.

Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 1419–1428, 2017.

Xinhang Li, Chong Chen, Xiangyu Zhao, Yong Zhang, and Chunxiao Xing. E4srec: An elegant effective efficient extensible solution of large language models for sequential recommendation. *arXiv preprint arXiv:2312.02443*, 2023.

Chengkai Liu, Jianghao Lin, Jianling Wang, Hanzhou Liu, and James Caverlee. Mamba4rec: Towards efficient sequential recommendation with selective state space models. *arXiv preprint arXiv:2403.03900*, 2024a.

Qidong Liu, Xian Wu, Yejing Wang, Zijian Zhang, Feng Tian, Yefeng Zheng, and Xiangyu Zhao. Llm-esr: Large language models enhancement for long-tailed sequential recommendation. *Advances in Neural Information Processing Systems*, 37:26701–26727, 2024b.

Khushhall Chandra Mahajan, Amey Porobo Dharwadker, Saurabh Gupta, and Brad Schumitsch. Videorecsys 2023: First workshop on large-scale video recommender systems. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pp. 1269–1271, 2023.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pp. 43–52, 2015.

Sören Mindermann, Jan M Brauner, Muhammed T Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N Gomez, Adrien Morisot, Sebastian Farquhar, et al. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *International Conference on Machine Learning*, pp. 15630–15649. PMLR, 2022.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems*, 36:10299–10315, 2023.

Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*, pp. 811–820, 2010.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*, 2012.

Alan Said, Brijnesh J Jain, Sascha Narr, and Till Plumbaum. Users and noise: The magic barrier of recommender systems. In *International conference on user modeling, adaptation, and personalization*, pp. 237–248. Springer, 2012.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015.

Yehjin Shin, Jeongwhan Choi, Hyowon Wi, and Noseong Park. An attentive inductive bias for sequential recommendation beyond the self-attention. In *Proceedings of the AAAI conference on artificial intelligence*, pp. 8984–8992, 2024.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pp. 1441–1450, 2019.

Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pp. 565–573, 2018.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. Contrastive learning for sequential recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*, pp. 1259–1273. IEEE, 2022.

Wujiang Xu, Qitian Wu, Zujie Liang, Jiaojiao Han, Xuying Ning, Yunxiao Shi, Wenfang Lin, and Yongfeng Zhang. Slmrec: Distilling large language models into small for sequential recommendation. *arXiv preprint arXiv:2405.17890*, 2024.

Hongzhi Yin, Bin Cui, Jing Li, Junjie Yao, and Chen Chen. Challenging the long tail recommendation. *arXiv preprint arXiv:1205.6700*, 2012.

Hengtong Zhang, Yaliang Li, Bolin Ding, and Jing Gao. Practical data poisoning attack against next-item recommendation. In *Proceedings of the web conference 2020*, pp. 2458–2464, 2020.

Weinan Zhang, Tianqi Chen, Jun Wang, and Yong Yu. Optimizing top-n collaborative filtering via dynamic negative item sampling. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pp. 785–788, 2013.

Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pp. 1893–1902, 2020.

Kun Zhou, Hui Yu, Wayne Xin Zhao, and Ji-Rong Wen. Filter-enhanced mlp is all you need for sequential recommendation. In *Proceedings of the ACM web conference 2022*, pp. 2388–2399, 2022.

Algorithm 1 Estimator-guided selective learning for sequential recommendation

Require: Training set $\mathcal{D}_{\text{train}}$, trainable backbone $f(\cdot;\theta)$, pre-trained estimator $g(\cdot;\phi)$, masking ratio ρ, learning rate η, number of epochs E
Ensure: Optimized backbone $f(\cdot;\theta)$

Freeze the estimator parameters ϕ
for $e = 1$ to E **do**
  for each mini-batch $\mathcal{B} = \{(x_i, y_i)\}_{i=1}^{B}$ sampled from $\mathcal{D}_{\text{train}}$ **do**
    Compute the backbone sample-wise losses:

$$\mathcal{L}_{\text{back}}^{i} = \ell(f(x_i;\theta), y_i), \quad i = 1, \ldots, B$$

    Compute the estimator sample-wise losses without gradient:

$$\mathcal{L}_{\text{est}}^{i} = \ell(g(x_i;\phi), y_i), \quad i = 1, \ldots, B$$

    Compute the estimator-guided score for each sample:

$$s_i = \mathcal{L}_{\text{back}}^{i} - \mathcal{L}_{\text{est}}^{i}$$

    Set the number of retained samples as

$$m = \max\left(\lfloor (1-\rho)B \rfloor, 1\right)$$

    Select the top-scored samples:

$$\mathcal{K} = \text{TopK}\left(\{s_i\}_{i=1}^{B}, m\right)$$

    Compute the final selective loss:

$$\mathcal{L}_{\text{final}} = \frac{1}{|\mathcal{K}|} \sum_{i \in \mathcal{K}} \mathcal{L}_{\text{back}}^{i}$$

    Update the backbone:

$$\theta \leftarrow \theta - \eta \nabla_{\theta} \mathcal{L}_{\text{final}}$$

  end for
end for
return $f(\cdot;\theta)$

A Additional Experimental Details

A.1 Training Procedure

The detailed training procedure of selective learning is summarized in Algorithm 1.

A.2 Datasets

We provide additional descriptions of the six benchmark datasets used in our experiments: Amazon Beauty, Amazon Sports and Outdoors, Amazon Musical Instruments, LastFM, MovieLens-1M, and Yelp. These datasets cover different recommendation domains, including e-commerce, business recommendation, movie recommendation, and music recommendation. Following previous studies (Kang & McAuley, 2018; Zhou et al., 2022), we apply 5-core filtering to all datasets, retaining only users and items with at least five interactions. The dataset statistics after preprocessing are reported in Table 1.

- **Amazon datasets.** Amazon Beauty, Amazon Sports and Outdoors, and Amazon Musical Instruments [2] are category-specific subsets of the Amazon product review corpus (McAuley et al., 2015). These datasets contain user–item interactions collected from product ratings and reviews on Amazon. They are widely used in sequential recommendation and are

[2] https://cseweb.ucsd.edu/~jmcauley/datasets.html#amazon_reviews

known for sparse interactions and relatively short sequence lengths. We use these three categories to evaluate the proposed method across different e-commerce domains.

- **LastFM.** LastFM [3] is a music recommendation dataset containing user interactions with music items, such as artist listening records. It is commonly used to recommend artists and music-related content to users. Since LastFM contains relatively long user interaction sequences, it provides a useful benchmark for evaluating sequential recommendation in the music domain.
- **MovieLens.** MovieLens-1M [4] is a popular benchmark dataset for movie recommendation. It contains about one million user ratings on movies. Compared with the other datasets used in our experiments, MovieLens-1M has relatively long user interaction sequences. We use this dataset to evaluate the proposed method in a movie recommendation setting.
- **Yelp.** Yelp [5] is a widely used business recommendation dataset. It contains user interactions with points of interest, such as check-in records and review information. In this work, we use Yelp to evaluate the proposed method in a business recommendation scenario, which differs from product recommendation datasets such as Amazon.

A.3 Baselines

This section provides additional descriptions of the four sequential recommendation backbones used in our experiments: SASRec, GRU4Rec, FMLPRec, and Mamba4Rec. These models cover different architectural families, including self-attention, recurrent neural networks, frequency-domain filtering, and state space models.

- **SASRec** (Kang & McAuley, 2018). SASRec is a representative Transformer-based sequential recommendation model. It adopts causal self-attention to capture dependencies among items in a user's historical interaction sequence while preserving the temporal order of interactions. By modeling both short-range and long-range item relations, SASRec has been widely used as a strong baseline for next-item recommendation.
- **GRU4Rec** (Hidasi et al., 2016). GRU4Rec is an RNN-based sequential recommendation model that uses gated recurrent units to encode user behavior sequences. It updates the hidden representation step by step as the interaction sequence evolves, making it suitable for capturing temporal dependencies in user behavior. GRU4Rec represents a classical recurrent architecture for sequential recommendation and is commonly used as a baseline in this field.
- **FMLPRec** (Zhou et al., 2022). FMLPRec is an MLP-based sequential recommendation model that replaces self-attention with learnable filters in the frequency domain. It models sequential patterns by transforming item representations into the frequency space and filtering noisy or less useful components. This design provides an efficient alternative to attention-based models for sequential recommendation.
- **Mamba4Rec** (Liu et al., 2024a). Mamba4Rec is a sequential recommendation model based on selective state space modeling. It is designed to capture long-range user behavior patterns without relying on quadratic self-attention. By using selective state space dynamics, Mamba4Rec provides an efficient backbone for modeling long interaction sequences in sequential recommendation.

A.4 Experimental Environment

All backbone models and their selective-learning variants were trained and evaluated on a server equipped with an AMD EPYC 7763 CPU and an NVIDIA A100 GPU with 80GB memory. The implementation was based on Ubuntu 22.04.4 LTS, Python 3.10.12, PyTorch 2.8.0, CUDA 12.8, and NVIDIA Driver 575.57.08.

[3] https://grouplens.org/datasets/hetrec-2011/
[4] https://grouplens.org/datasets/movielens/
[5] https://www.yelp.com/dataset

B More Experimental Results

B.1 Loss Curve Analysis

We provide additional training loss curves for FMLPRec and Mamba4Rec on Sports, LastFM, and Yelp in Figure 6. Across both backbones, SL exhibits steadily decreasing loss curves, although its selective loss remains higher than EST in most cases. This is consistent with the design of selective learning, where the final objective is computed only over retained high-score instances rather than averaged over all training instances. Therefore, a higher SL loss reflects the selected subset used for optimization rather than unstable training. Although EST often reduces the training loss faster than SL, lower training loss does not necessarily imply better recommendation performance. Since EST optimizes all training instances uniformly, it may also fit weakly informative or less generalizable behavioral contexts. SL instead filters low-score samples and updates the backbone using retained high-score instances, which can improve generalization even when the selective loss remains higher. The absolute loss scale differs across backbones, especially for Mamba4Rec, so the curves should be interpreted within each architecture instead of being compared directly across models. Overall, these results suggest that selective learning changes the composition of samples contributing to the update while maintaining stable convergence, providing additional evidence for the proposed training mechanism.

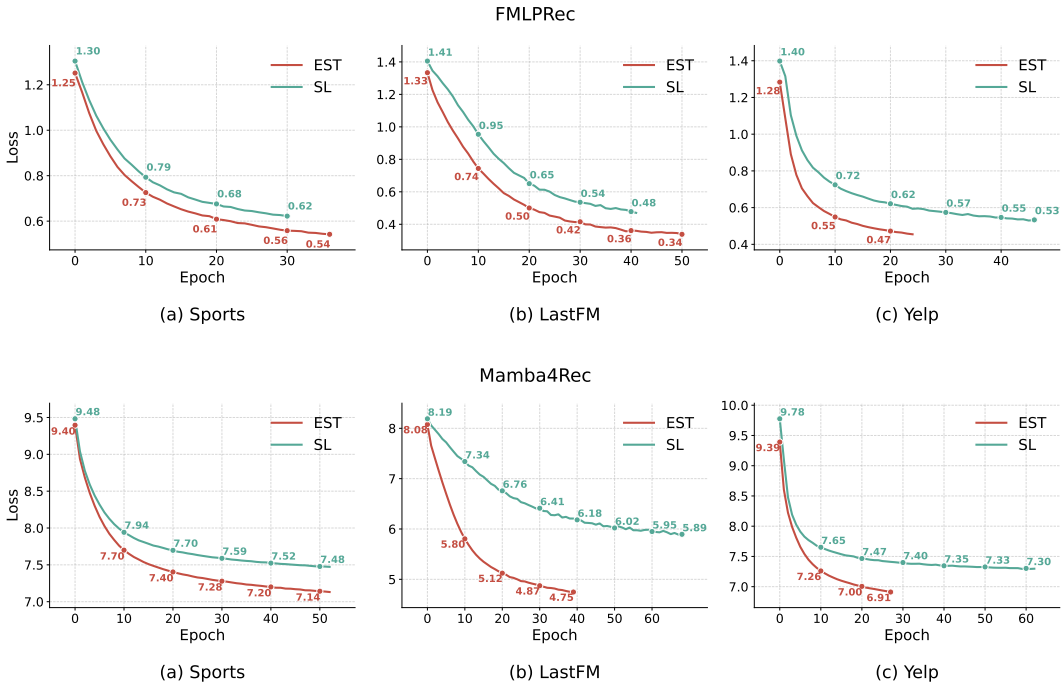
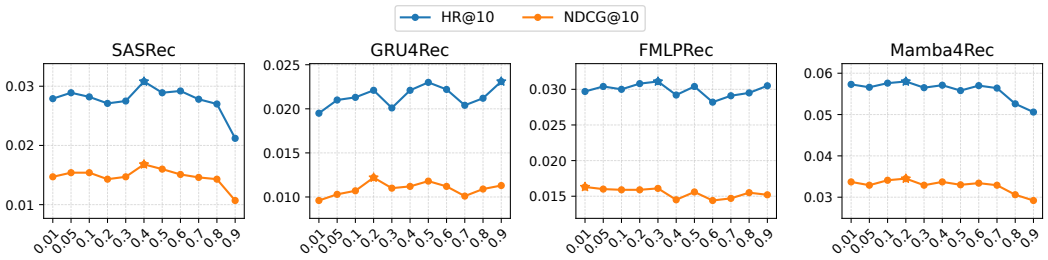


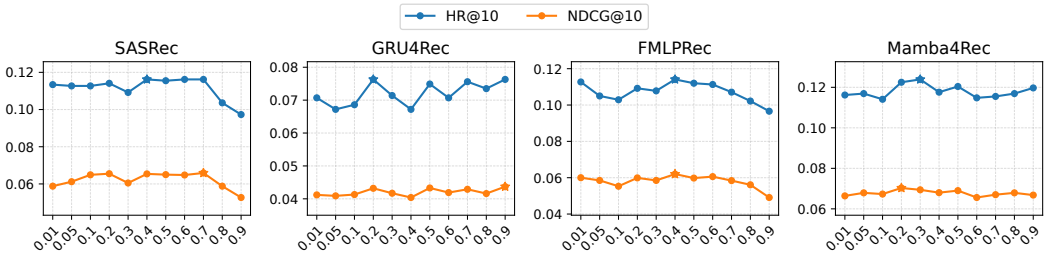
Figure 6: Training loss curves of FMLPRec and Mamba4Rec under standard and selective learning on Sports, LastFM, and Yelp.

B.2 Selective Mask Ratio Analysis

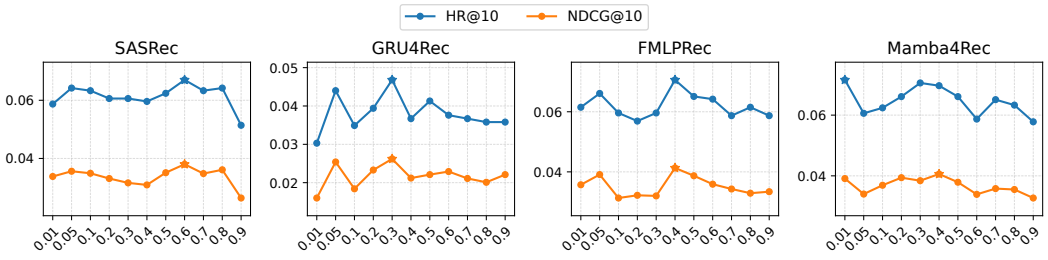
We provide the full results of the selective mask ratio analysis in Figure 7. As discussed in the main text, the optimal masking ratio varies across datasets and backbone architectures, indicating that the appropriate degree of selective learning depends on both data characteristics and model-specific learning dynamics. For reproducibility, the best masking ratios used for each dataset and backbone are summarized in Table 5.



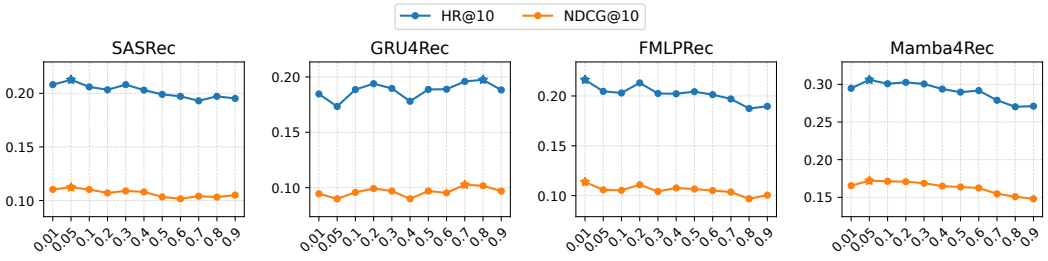
(a) Sports



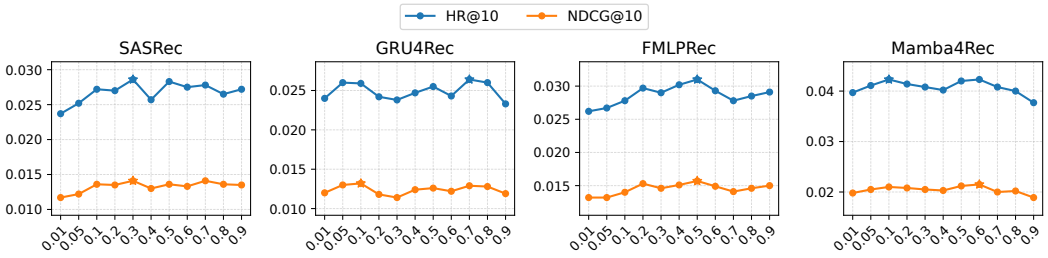
(b) Instruments



(c) LastFM



(d) ML-1M

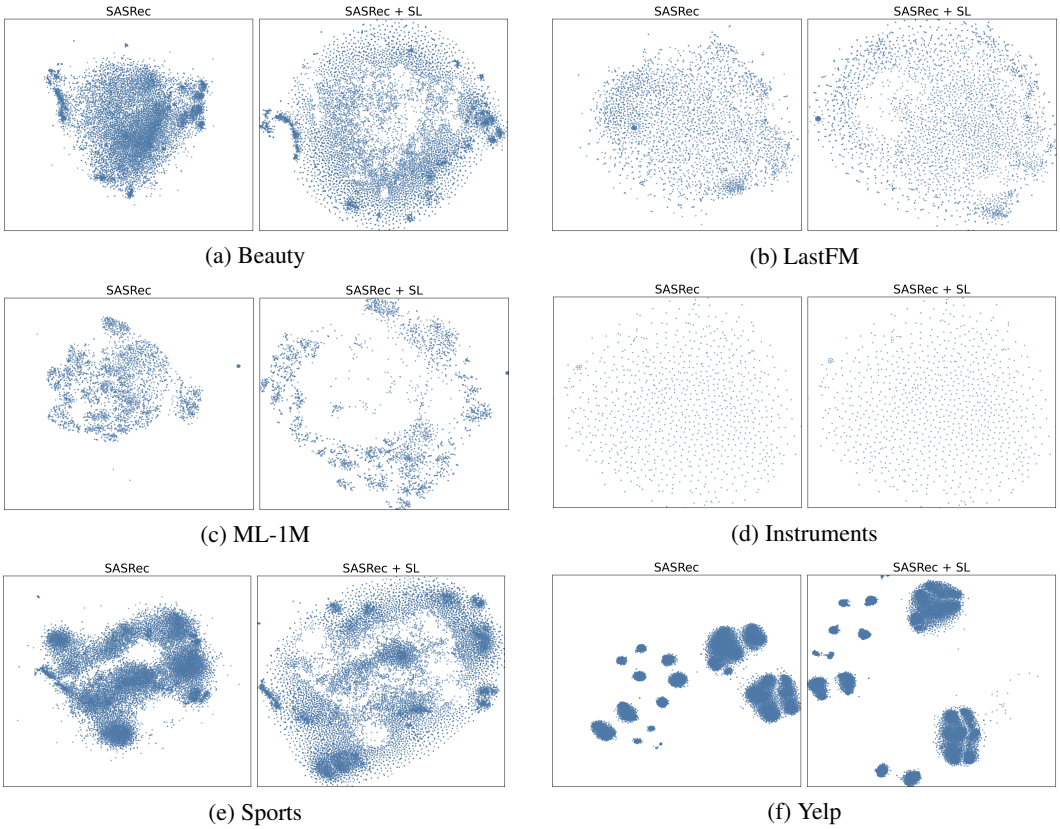


(e) Yelp

Figure 7: Impact of selective mask ratio across the remaining datasets.

Table 5: Best selective mask ratios across datasets and backbones.

Datasets	SASRec + SL	GRU4Rec + SL	FMLPRec + SL	Mamba4Rec + SL
Beauty	0.3	0.2	0.1	0.4
Sports	0.4	0.2	0.01	0.2
Instruments	0.5	0.7	0.3	0.2
LastFM	0.6	0.3	0.4	0.8
ML-1M	0.05	0.7	0.01	0.2
Yelp	0.3	0.7	0.5	0.5



(a) Beauty (b) LastFM

(c) ML-1M (d) Instruments

(e) Sports (f) Yelp

Figure 8: t-SNE visualizations of item embeddings learned by SASRec and SASRec with SL on all datasets.

B.3 Embedding Space Visualization

To qualitatively examine the effect of selective learning on learned item representations, we visualize the item embedding space of SASRec across all datasets using t-SNE (Van der Maaten & Hinton, 2008). As shown in Figure 8, the embeddings learned with SL generally exhibit less concentrated and more broadly distributed structures than those learned by the original backbone, although the extent of change varies across datasets. On Beauty, the original embeddings are concentrated within a relatively compact region, whereas those learned with SL are more widely dispersed in the projected space. More noticeable changes are also observed on LastFM, ML-1M, and Sports, while Yelp exhibits more localized refinements in its cluster structure. For Instruments, where the embeddings are already broadly distributed, only minor changes are observed. These observations are consistent with the training mechanism of selective learning. Since SL prioritizes instances with larger estimator-guided loss discrepancies, the backbone is updated using samples that remain relatively underfitted with respect to the reference estimator. Such selective updates may expose the model to more diverse optimization signals and thereby alter the geometry of the learned represen-

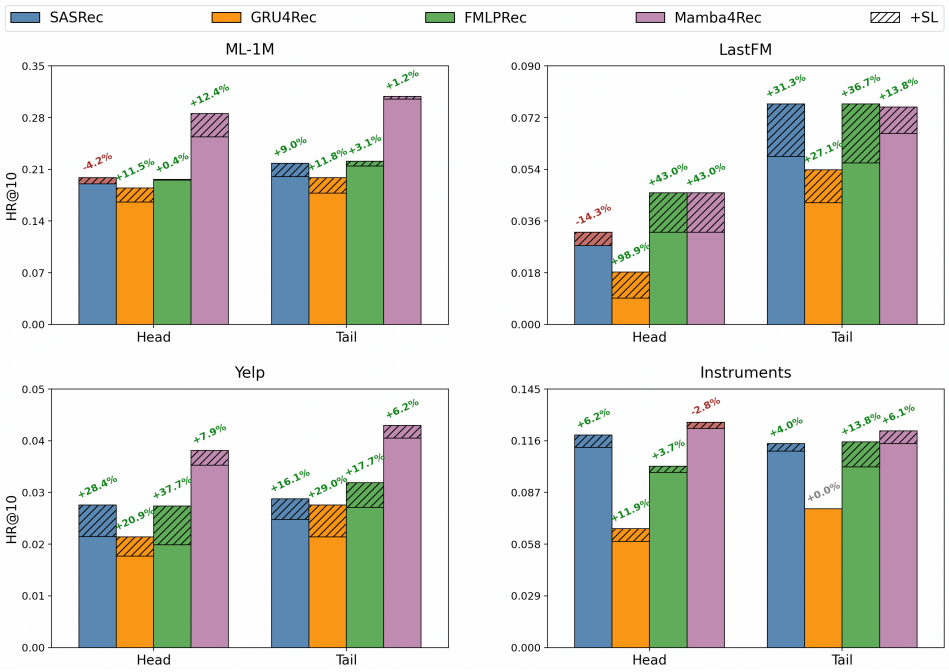


Figure 9: Performance comparison across head and tail user groups on the remaining datasets.

tation space. We emphasize that these visualizations serve as qualitative supporting evidence rather than a standalone explanation of the performance gains.

B.4 Long-tailed Recommendation

To further examine the behavior of selective learning under long-tailed settings, we provide additional results on ML-1M, LastFM, Yelp, and Instruments in Figure 9. The overall tendency is consistent with the observations in the main text. Selective learning generally improves or maintains performance across both head and tail user groups, while also yielding clear improvements for tail users where interaction signals are relatively limited. Across datasets, the magnitude of improvement varies, but no consistent degradation is observed. Minor performance decreases occur in some cases; however, these changes are not systematic and do not affect the overall trend. In particular, more noticeable improvements appear in relatively sparse datasets (e.g., LastFM and Yelp), whereas more moderate but consistent changes are observed in denser datasets such as ML-1M and Instruments. These results indicate that the effectiveness of selective learning is not limited to specific datasets, but extends across different data regimes. These observations are aligned with the design of selective learning, which emphasizes informative training instances during optimization. As a result, the model can more effectively leverage limited supervision signals in tail regimes while maintaining stable performance in data-rich settings.

B.5 Efficiency Analysis

The proposed selective learning framework introduces no additional inference cost compared with the original backbone models, since the pre-trained estimator is used only during training and discarded at inference time. Therefore, we analyze the efficiency of SL by measuring the training runtime per epoch. As shown in Figure 10, applying SL slightly increases the training runtime due to the additional estimator forward pass and sample selection process. However, this moderate training overhead is accompanied by consistent performance improvements across datasets and backbone models. These results indicate that SL provides a favorable trade-off between training cost and recommendation accuracy.

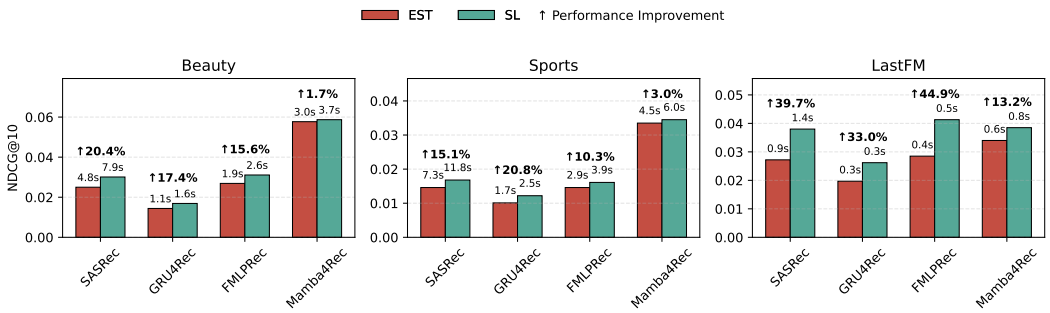


Figure 10: Training runtime (seconds) per epoch and performance comparison with and without selective learning.

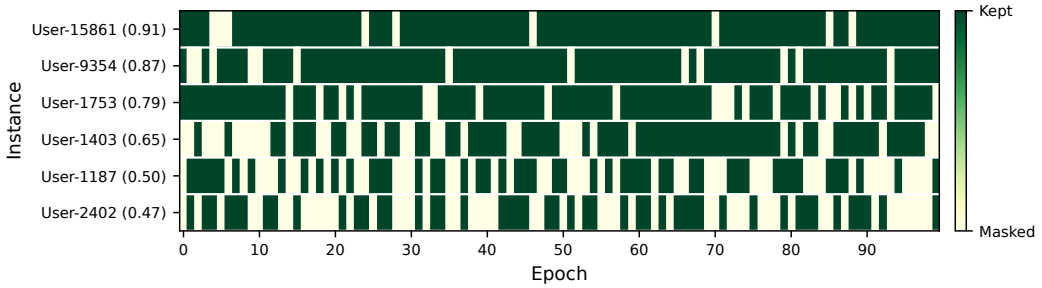


Figure 11: Epoch-wise selection decisions for representative training instances on the Beauty dataset with SASRec as the backbone. Each cell indicates whether the instance is retained or masked at a given epoch, as shown by the color bar. The y-axis reports the user ID and selection rate, i.e., the fraction of epochs in which the instance is retained.

B.6 Case Study

To better understand how selective learning behaves during training, we visualize epoch-wise selection decisions for representative training instances. As shown in Figure 11, the retained instances are not fixed throughout training. Some instances are selected in most epochs, while others are selected only intermittently or change their selection status as training progresses. This indicates that selective learning is not a static data filtering strategy, but a dynamic training mechanism that adjusts which instances contribute to the update according to the estimator-guided score.

B.7 Understanding Instance Scores

To provide an intuitive understanding of what the proposed instance score captures, we qualitatively examine representative high-gap training instances from ML-1M. For interpretability, item-level interactions are represented by their movie genres. Table 6 reports the input history, target genre, subsequent interactions, loss gap, and the target ranks produced by the backbone and estimator. A lower target rank indicates that the model assigns higher preference to the ground-truth target item. Across all three cases, the backbone assigns the target a substantially worse rank than the estimator, resulting in large loss gaps. Importantly, these targets do not appear to be arbitrary or clearly noisy interactions. In Case 2, for example, the target is Drama and is followed by three consecutive Drama interactions, suggesting that the target is consistent with the user's subsequent preference pattern. Similarly, in Case 3, the Thriller target is immediately followed by another Thriller interaction. Nevertheless, the backbone assigns substantially worse ranks to these targets than the estimator. These observations provide an intuitive interpretation of the proposed score. A large loss gap does not necessarily indicate that an instance is intrinsically difficult or noisy; instead, it identifies an instance for which the current backbone exhibits a larger fit discrepancy relative to the fixed estimator. Such instances can therefore provide informative optimization signals that have

Table 6: Representative high-gap training instances. Lower target rank is better.

Case	Input history	Target	Subsequent interactions	Loss gap	Backbone rank	Estimator rank
1	Film-Noir→Film-Noir	Drama	Comedy→Drama→Action	3.884	1847	550
2	Thriller→Romance	Drama	Drama→Drama→Drama	4.332	1910	934
3	Action→Animation	Thriller	Thriller→Comedy→Horror	4.469	2169	1324

Table 7: Performance comparison of score-based selection strategies on the Beauty dataset. All Top, Middle, and Bottom tier variants retain the same proportion of training instances.

Backbone	Selection strategy	HR@10	NDCG@10
SASRec	Top 10%	0.0449	0.0225
	Middle 10%	0.0067	0.0042
	Bottom 10%	0.0047	0.0018
	Top 10% w/o Top-1	**0.0476**	**0.0238**
	Top-1	0.0109	0.0051
FMLPRec	Top 10%	**0.0502**	**0.0248**
	Middle 10%	0.0053	0.0022
	Bottom 10%	0.0046	0.0026
	Top 10% w/o Top-1	0.0492	0.0246
	Top-1	0.0097	0.0051

not yet been sufficiently captured by the current model, illustrating how estimator-guided scoring can identify training instances that remain relatively under-fitted by the backbone.

B.8 Analysis of Score Tiers

To assess whether the proposed instance score provides a meaningful signal for sample selection, we compare different score tiers under the same retained-sample budget. Specifically, within each mini-batch, we rank training instances according to their loss-gap scores and independently retain the Top 10% (i.e., the highest-scored instances), Middle 10%, or Bottom 10% for optimization. All three variants therefore use the same number of training instances, allowing us to isolate the effect of score-based ranking from the effect of simply reducing the number of samples. As shown in Table 7, selecting the Top 10% consistently yields substantially better performance than selecting the Middle or Bottom 10% for both SASRec and FMLPRec. These results indicate that the proposed score provides a meaningful ranking signal for identifying training instances that are more beneficial to the current optimization process, beyond the generic effect of reducing the number of participating samples. We further examine whether the single highest-scored instance is responsible for the behavior of the top tier. Removing the Top-1 instance from the Top 10% slightly improves performance for SASRec but slightly decreases it for FMLPRec. This inconsistent effect suggests that the highest-scored instance is not systematically detrimental across backbone models. In contrast, training exclusively with the Top-1 instance results in substantial performance degradation for both backbones, indicating that relying on a single instance per update provides insufficient training signals despite its high score.