

Divergence Meets Consensus: A Multi-Source Negative Sampling Framework for Sequential Recommendation

Yuanzi Li Lingjie Wang Jingyu Zhao Zihang Tian
Renmin University of China Shandong University Shandong University Renmin University of China
Beijing, China Qingdao, China Qingdao, China Beijing, China
liyuanzi0313@outlook.com lingjie.wang@mail.sdu.edu.cn jingyu.zhao@mail.sdu.edu.cn zihangtian@ruc.edu.cn

Yuhan Wang Lei Wang Xu Chen*
Peking University Renmin University of China Renmin University of China
Beijing, China Beijing, China Beijing, China
fzzh040114@gmail.com wanglei154@ruc.edu.cn xu.chen@ruc.edu.cn

Abstract

Negative sampling is significant for training sequential recommendation models under implicit feedback. The predominant strategy, self-guided hard negative sampling, selects negatives based on the model's current state but suffers from three limitations: (1) the coupling between sampling and model updates triggers a vicious cycle that drives the model into local optima; (2) relying on current model parameters narrows sampling to a small region of the item space, reducing diversity and harming generalization; (3) identifying a hard negative requires scoring the entire candidate pool, causing substantial computational overhead with minimal information gain.

To address these challenges, we propose **MDCNS** (**M**ulti-source **D**ivergence-**C**onsensus for **N**egative **S**ampling), a novel "Teacher-Peer-Self" framework inspired by Vygotsky's Zone of Proximal Development (ZPD) theory. The proposed method comprises three components, including *multi-source scoring*, *divergence re-ranking*, and *consensus distillation*. Firstly, multi-source scoring incorporates peer and ensemble teacher models to inject external negative signals and break the self-reinforcement loop. Then, divergence re-ranking exploits prediction discrepancy between self and peer models to enhance sampling diversity. Finally, consensus distillation aligns the self model with the teacher via KL divergence, simultaneously improving computational cost utilization. Extensive experiments on six real-world datasets and five backbone models show that MDCNS consistently outperforms state-of-the-art negative sampling methods, demonstrating strong effectiveness and generalization. Code is available at <https://github.com/Lyz103/SIGIR26-MDCNS>.

CCS Concepts

• **Information systems** → **Information retrieval**; **Recommender systems**.

Keywords

Sequential Recommendation; Negative Sampling

*Corresponding author



This work is licensed under a Creative Commons Attribution 4.0 International License. *SIGIR '26, Melbourne, VIC, Australia*

© 2026 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2599-9/2026/07

<https://doi.org/10.1145/3805712.3809555>

ACM Reference Format:

Yuanzi Li, Lingjie Wang, Jingyu Zhao, Zihang Tian, Yuhan Wang, Lei Wang, and Xu Chen. 2026. Divergence Meets Consensus: A Multi-Source Negative Sampling Framework for Sequential Recommendation. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '26)*, July 20–24, 2026, Melbourne, VIC, Australia. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3805712.3809555>

1 Introduction

Sequential Recommendation. Recommender systems have become a cornerstone technology in modern information access, and are now deeply integrated into a wide range of application scenarios, such as e-commerce [13], social media [42], and other online services and platforms [23]. Among the diverse recommendation paradigms, sequential recommendation (SR) has recently emerged as a central research topic, drawing sustained interest from both academic and industrial communities. The fundamental goal of SR is to exploit users' historical interaction sequences to infer and anticipate their upcoming preferences and potential future behaviors. In most real-world systems, SR models are trained on implicit feedback (e.g., clicks and views), which makes it crucial to construct reliable sets of positive and negative instances for effective learning. Because implicit feedback predominantly reflects positive user signals, negative sampling becomes a key mechanism for creating contrastive supervision that enables discriminative training. Conceptually, existing negative sampling strategies can be divided into two families with different levels of modeling sophistication:

Level 0 (L_0): Heuristic-based Negative Sampling. This level relies on static, data-driven heuristics independent of the model's training state. Existing heuristic rules typically generate negatives by either treating all unseen items equally [25], leveraging exposure logs to identify unclicked candidates [5], or biasing the sampling distribution according to item popularity [3, 43]. While highly efficient, these Level 0 strategies primarily produce uninformative negatives. This lack of robust contrastive signals eventually causes gradient vanishing, limiting the model's capacity to learn nuanced preference boundaries under complex data distributions.

Level 1 (L_1): Self-guided Hard Negative Sampling. To ensure effective parameter updates and compel the model to learn more discriminative user preference representations, hard negative samples are employed—instances difficult to distinguish from positives (e.g., for a positive smartphone, a phone case is more informative

Table 1: Characteristics comparison between MDCNS and existing methods. The abbreviations denote: SGS (Self-Guided Sampling), MSS (Multi-Source Scoring), and ERA (Efficient Resource Allocation). In the PRF column, a higher number of ○ marks indicates superior performance.

Level	Method	SGS	MSS	ERA	PRF
L ₀	RNS [25]	✗	✗	✓	○
	PNS [3]	✗	✗	✓	○
	ENS [5]	✗	✗	✓	○
L ₁	DNS [44]	✓	✗	✗	○○
	MixGCF [15]	✓	✗	✗	○○
	AdaSIR [1]	✓	✗	✗	○○
	GNNO [7]	✓	✗	✗	○○
	DNS+ [29]	✓	✗	✗	○○
	SRNS [6]	✓	✗	✗	○○
L ₂	MDCNS	✓	✓	✓	○○○

than a frying pan). To identify hard negative samples, it is standard practice to leverage a *self-guided* sampling mechanism [14, 44], the core logic of which is to identify challenging negative samples in real-time by utilizing the model’s *current* parameter state. In practice, this mechanism first constructs a randomly sampled candidate pool, then scores the candidates within this pool based on the active training step’s parameters, and finally selects a *single* top-ranked item as the hard negative [1, 29]. This approach ensures a dynamic alignment between negative sample hardness and the model’s evolving predictive capability, thereby facilitating more effective and discriminative parameter updates.

Remaining Limitations. Although L₁ methods have achieved certain progress, there still exist the following limitations:

(1) *Negative self-reinforcement loop.* Self-guided hard negative sampling creates a tight coupling between the sampling strategy and the model’s current state. This dependency risks triggering a vicious cycle: if the model selects bad negatives, the subsequent gradient updates will degrade the model’s discriminative ability. This degradation, in turn, yields even lower-quality samples in future iterations, thereby trapping the model in a local optimum.

(2) *Restricted sampling diversity.* Relying solely on current model parameters for negative sample construction confines the mechanism to select negatives solely within the local item space currently emphasized by the model, failing to effectively explore the broader item spectrum. Consequently, the sampling distribution progressively shrinks, making it difficult to sustain or enhance sample diversity across successive training iterations, thereby severely constraining the model’s overall generalization capability.

(3) *Inefficient resource allocation.* The mechanism necessitates scoring the entire pool of potential candidates to identify a single hard negative, leading to significant computational redundancy. The vast majority of the computed scores are discarded without contributing to the gradient update, resulting in a disproportionately high computational overhead relative to the information gain, which severely reduces the overall training efficiency.

Our Proposed Framework. To transcend the inherent limitations of L₁-level self-learning, we draw inspiration from Vygotsky’s Zone

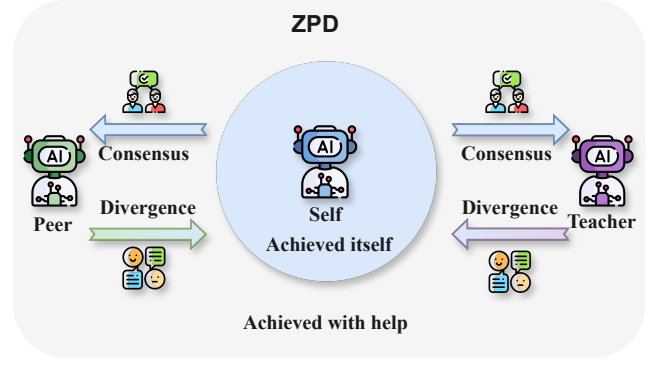


Figure 1: Illustration of the ZPD. It utilizes a "Teacher-Peer-Self" collaborative structure with Divergence and Consensus, thereby extending the learner’s capabilities from the inner self-achieved zone to the broader potential zone.

of Proximal Development (ZPD) theory [27]. As illustrated in Figure 1, ZPD posits that optimal learning emerges not in isolation, but within a collaborative triad of "Teacher-Peer-Self". We argue that the engine driving this progress is the dynamic interplay between expanding boundaries (via divergence) and consolidating knowledge (via consensus). Translating this educational philosophy into the realm of negative sampling, we propose **MDCNS** (Multi-source Divergence-Consensus for Negative Sampling). This framework is structurally designed to mirror the ZPD dynamic: it orchestrates multi-source negative samples to maintain a synergy between divergence and consensus, thereby systematically breaking the bottlenecks of isolated sampling methods.

(1) *Multi-source scoring.* To mitigate the negative self-reinforcement loop, instead of relying solely on the self model, we incorporate a peer model and an ensemble teacher model (derived from the former two) to get candidate items’ relevance scores independently. This mechanism disrupts the closed feedback loop, enabling the model to escape suboptimal states and enhancing overall performance.

(2) *Divergence re-ranking.* To address insufficient sampling diversity, we quantify the score discrepancy between the self model and the peer model across the candidate pool as a divergence score. This term is incorporated into the item relevance scores to re-rank the candidate items. We then randomly select from the top- M re-ranked items to ensure both diversity and appropriate difficulty.

(3) *Consensus distillation.* To address the problem of inefficient resource allocation, we use the KL divergence [32] to align the score distribution of the self model with that of the teacher model over the candidate item pool. This process facilitates the distillation of consensus knowledge from the teacher to the student (self model), simultaneously maximizing the utilization of training signals.

Table 1 shows the comprehensive comparison of key characteristics between MDCNS and the latest methods.

Contributions. Our contributions are summarized as follows:

- *Problem Identification.* We identify three critical limitations in existing L₁ negative sampling methods, which can be summarized as: the negative self-reinforcement loop, restricted sampling diversity, and inefficient resource allocation.

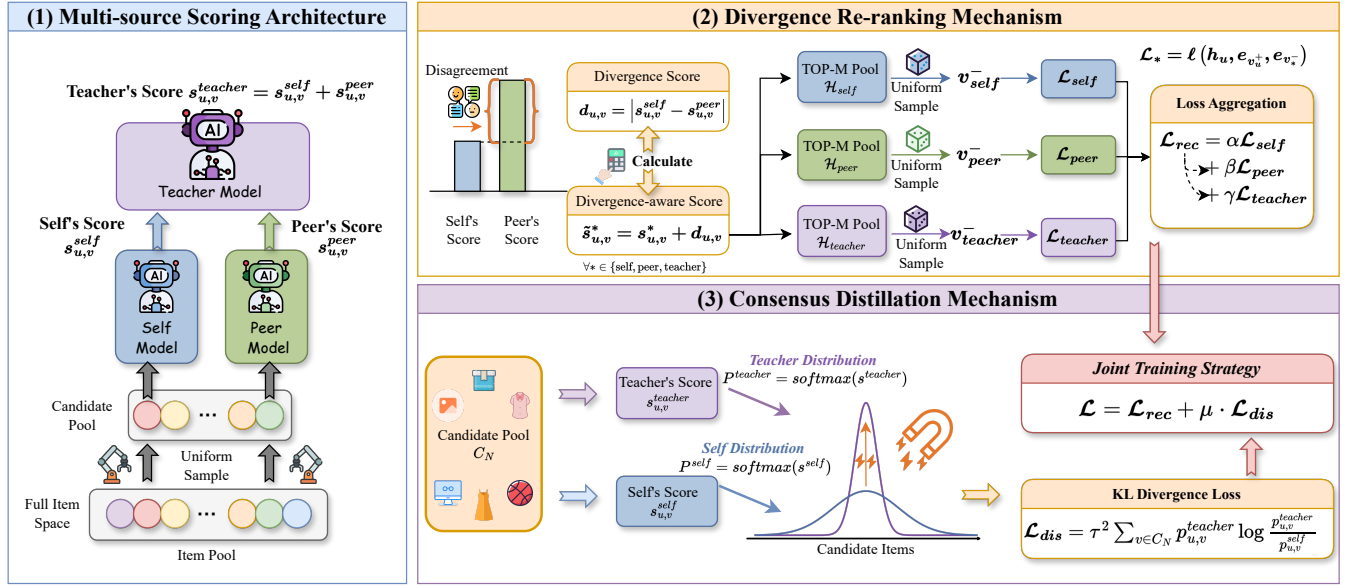


Figure 2: The overall framework of MDCNS. (1) Multi-source Scoring, which constructs a candidate pool and generates relevance scoring signals from Self, Peer, and Teacher perspectives; (2) Divergence Re-ranking, which quantifies view divergence scores between self model and peer model to re-rank candidates and samples diverse hard negatives from the Top-M list; and (3) Consensus Distillation, which utilizes the teacher’s distribution over the entire candidate pool as soft targets to guide the self model, effectively acquiring consensus knowledge while mitigating the inefficiencies of resource allocation.

- *Theoretical Perspective.* We analyze the bottlenecks of L_1 sampling through the lens of Vygotsky’s Zone of Proximal Development (ZPD) theory, proposing a shift from isolated self-guided to a collaborative “Teacher-Peer-Self” paradigm.
- *Methodological Framework.* We propose MDCNS, a ZPD-inspired framework that technically realizes the “Teacher-Peer-Self” structure via three core components: multi-source scoring, divergence re-ranking, and consensus distillation.
- *Experimental Validation.* Extensive experiments on various real-world datasets, backbones and loss functions demonstrate that MDCNS consistently outperforms state-of-the-art baselines by a significant margin in both effectiveness and generalization.

2 Problem Formulation

Let $\mathcal{U}$ and $\mathcal{V}$ denote the sets of users and items, respectively. We represent the interaction history of a user $u \in \mathcal{U}$ as a chronologically ordered sequence $S_u = [v_1, v_2, \dots, v_{|S_u|}]$, where $v_t \in \mathcal{V}$ indicates the item interacted with at time step t .

The primary objective of sequential recommendation is to forecast the user’s next interaction at step $|S_u| + 1$. Formally, we aim to identify the candidate item v^* that maximizes the conditional probability given the user’s history:

$$\arg \max_{v^* \in \mathcal{V}} P(v_{|S_u|+1} = v^* | S_u). \quad (1)$$

To optimize for this objective, state-of-the-art methods typically employ either a pairwise ranking loss, such as bayesian personalized ranking $\mathcal{L}_{bpr}$, or a binary cross-entropy loss $\mathcal{L}_{bce}$. Specifically, $\mathcal{L}_{bpr}$ is designed to learn a scoring function that ranks the ground-truth next item (positive sample, v^+) higher than unobserved items

(negative samples, v^-). Conversely, $\mathcal{L}_{bce}$ formulates the recommendation task as a binary classification problem, distinguishing positive instances v^+ from sampled negative instances v^- .

3 Methodology

In this section, motivated by the Zone of Proximal Development (ZPD) theory, we propose MDCNS, a novel “Teacher-Peer-Self” negative sampling framework for sequential recommendation. Specifically, MDCNS incorporates three synergistic components: multi-source scoring, divergence re-ranking, and consensus distillation.

As illustrated in Figure 2, MDCNS first employs a multi-source scoring phase (Section 3.1) to derive candidate item relevance scores from the distinct perspectives of the teacher, peer, and self. Then, in the divergence re-ranking phase (Section 3.2), the divergence score is computed between the predictions of the peer and self models. Subsequently, the candidates are re-ranked by jointly considering both the item relevance scores and the divergence scores. Furthermore, a consensus distillation module (Section 3.3) is proposed to fully exploit the computational cost and distill the consensus knowledge from the teacher model into the self model. Finally, the time complexity of the proposed MDCNS is analyzed in Section 3.4.

3.1 Multi-source Scoring

Formally, let $S_u = [v_1, v_2, \dots, v_{|S_u|}]$ denote the interaction sequence for user u , where v_t represents the item interacted at step t . In our framework, we denote the self model to be trained as M_{self} (e.g., SASRec) and the peer model as M_{peer} (e.g., Mamba4Rec)¹, and the teacher model as $M_{teacher}$.

¹ M_{self} and M_{peer} function as mutual peers within a dual framework. M_{peer} follows an identical training procedure to M_{self} , we omit redundant descriptions for brevity.

3.1.1 Candidate Pool Construction. Given the computational intractability of performing full-ranking over the vast item space $\mathcal{V}$, we adopt a candidate generation strategy. Specifically, we construct a candidate set C_N by uniformly sampling N items from the unobserved item space (i.e., excluding the user's history S_u and the ground-truth target v_u^+). Formally, C_N is obtained via:

$$C_N \sim \text{Uniform}(\mathcal{V} \setminus (\{v_u^+\} \cup S_u), N). \quad (2)$$

3.1.2 Dual-View Scoring. Both models encode the input sequence S_u into latent representations. Let h_u^{self} and h_u^{peer} denote the sequence representations generated by M_{self} and M_{peer} , respectively. For every candidate item $v \in C_N$, let e_v be its item embedding. We compute the relevance scores by performing the dot product between the sequence representation and the item embeddings:

$$s_{u,v}^{\text{self}} = h_u^{\text{self}} \cdot e_v, \quad s_{u,v}^{\text{peer}} = h_u^{\text{peer}} \cdot e_v, \quad (3)$$

where $s_{u,v}^{\text{self}}$ and $s_{u,v}^{\text{peer}}$ represent the prediction scores for item v under the self and peer views, respectively. Consequently, we obtain the score sets for the candidate items: $S_{C_N}^{\text{self}} = \{s_{u,v}^{\text{self}} \mid v \in C_N\}$ and $S_{C_N}^{\text{peer}} = \{s_{u,v}^{\text{peer}} \mid v \in C_N\}$.

3.1.3 Teacher Signal Aggregation. To mitigate the bias inherent in a single model and leverage the consensus knowledge, we define the teacher model M_{teacher} as an additive aggregation of the two student models. The teacher's score for each candidate item $v \in C_N$ is calculated as:

$$s_{u,v}^{\text{teacher}} = s_{u,v}^{\text{self}} + s_{u,v}^{\text{peer}}. \quad (4)$$

We utilize $S_{C_N}^{\text{teacher}}$ for two distinct purposes: guiding negative sampling from the teacher's perspective and serving as soft targets for distilling knowledge into the self model.

3.2 Divergence Re-ranking

To alleviate the issue of restricted sampling diversity, we propose a divergence-based re-ranking strategy. This approach exploits the disagreement between the self and peer views to identify samples that are not only hard but also diverse.

3.2.1 Divergence Quantification and Re-scoring. We hypothesize that items with high scoring discrepancies between the self and peer models represent uncertain samples. For each candidate item $v \in C_N$, we quantify this disagreement as the absolute difference between their prediction scores, termed the Divergence Score $d_{u,v}$:

$$d_{u,v} = |s_{u,v}^{\text{self}} - s_{u,v}^{\text{peer}}|. \quad (5)$$

To prioritize these informative samples, we augment the original prediction scores with the divergence term. Specifically, we update the scores for the self, peer, and teacher views as follows:

$$\tilde{s}_{u,v}^* = s_{u,v}^* + d_{u,v}, \quad \forall * \in \{\text{self}, \text{peer}, \text{teacher}\} \quad (6)$$

where $\tilde{s}_{u,v}^*$ denotes the divergence-aware score. This formulation ensures that items with both high relevance probability and high view disagreement are ranked higher.

Algorithm 1: The training process of MDCNS

Input: Interaction sequences, candidate size N

Output: Learned parameter Θ of M_{self}

```

1 for epoch  $\leftarrow 1$  to  $T$  do
2   foreach sequence  $S_u$  do
3     // Multi-source Scoring
4     Construct candidate set  $C_N$  according to Eq. (2)
5     Obtain  $s^{\text{self}}, s^{\text{peer}}$  and  $s^{\text{teacher}}$  for  $C_N$  (Eqs. (3)–(4))
6     // Divergence Re-ranking
7     Get divergence-aware scores  $\tilde{s}$  using Eq. (5) to Eq. (6)
8     foreach view  $*$   $\in \{\text{self}, \text{peer}, \text{teacher}\}$  do
9       | Select  $v_*^-$  from Top-M candidates (Eqs. (7)–(8))
10    end
11    Calculate ranking loss  $\mathcal{L}_{\text{rec}}$  via Eqs. (9)–(10)
12    // Consensus Distillation
13    Calculate consensus loss  $\mathcal{L}_{\text{dis}}$  via Eqs. (11)–(12)
14    Update  $\Theta$  by minimizing  $\mathcal{L}_{\text{rec}} + \mu \mathcal{L}_{\text{dis}}$ 
15  end
16 end
17 return  $M_{\text{self}}$  with  $\Theta$ 

```

3.2.2 Hard Negative Sampling. To construct robust negative sets, we first identify the most informative candidates for each view. Specifically, for each perspective $* \in \{\text{self}, \text{peer}, \text{teacher}\}$, we select the top- M items from the candidate pool C_N based on their divergence-aware scores $\{\tilde{s}_{u,k}^*\}$:

$$\mathcal{H}_* = \text{Top-M}(\{\tilde{s}_{u,k}^* \mid k \in C_N\}). \quad (7)$$

Subsequently, to mitigate the false negative problem inherent in strict hard mining and to introduce stochasticity for robust training, we uniformly sample one negative instance from each top-ranked pool:

$$v_*^- \sim \text{Uniform}(\mathcal{H}_*). \quad (8)$$

In this manner, the negative samples are selected to ensure that they are not only challenging but also diverse.

3.2.3 Recommendation Optimization Objective. To train the Self-Model M_{self} , we employ a standard ranking loss $\ell(\cdot)$ (e.g., BPR or BCE) and formulate the view-specific optimization objective as:

$$\mathcal{L}^* = \ell(h_u^{\text{self}}, e_{v_u^+}, e_{v_*^-}), \quad (9)$$

where h_u^{self} is the user representation from the self-model, $e_{v_u^+}$ is the embedding of the ground-truth positive item, and $e_{v_*^-}$ is the embedding of the negative item sampled from the perspective $* \in \{\text{self}, \text{peer}, \text{teacher}\}$. Finally, the total recommendation objective is computed as a weighted aggregation of the losses derived from the three distinct negative sampling distributions:

$$\mathcal{L}_{\text{rec}} = \alpha \mathcal{L}_{\text{self}} + \beta \mathcal{L}_{\text{peer}} + \gamma \mathcal{L}_{\text{teacher}}, \quad (10)$$

where α, β , and γ are hyperparameters that balance the contribution of hard negatives mined from the views of the self, the peer and the teacher, respectively.

3.3 Consensus Distillation

The previous stage incurs the computational cost of scoring the candidate set C_N , yet standard sampling exploits only one instance. To maximize the utility of these computations, we propose a consensus distillation module. By distilling the distribution of $M_{teacher}$ over C_N into M_{self} , we preserve the global ranking signals that are otherwise discarded by hard sampling.

3.3.1 Probabilistic Soft Labeling. To align the granular ranking views of the teacher and self-model, we transform their raw scores into probability distributions over the candidate set C_N . Specifically, we employ a temperature-scaled Softmax function to normalize the scores. For a user u and item $v \in C_N$, the probability under view $* \in \{\text{self}, \text{teacher}\}$ is computed as:

$$p_{u,v}^* = \frac{\exp(s_{u,v}^*/\tau)}{\sum_{j \in C_N} \exp(s_{u,j}^*/\tau)}, \quad (11)$$

where τ is a temperature hyperparameter regulating the sharpness of the distribution.

3.3.2 Distillation Objective. We adopt the KL Divergence [32] to minimize the discrepancy between the teacher’s consensus distribution and the self-model’s predicted distribution. Following the standard knowledge distillation paradigm [12], we use the temperature τ as the scaling factor to control the gradient magnitudes. The distillation loss is formulated as:

$$\mathcal{L}_{dis} = \tau^2 \sum_{v \in C_N} p_{u,v}^{teacher} \log \frac{p_{u,v}^{teacher}}{p_{u,v}^{self}}. \quad (12)$$

By minimizing $\mathcal{L}_{dis}$, M_{self} is encouraged to approximate the ensemble prediction $M_{teacher}$, thereby implicitly incorporating the teacher knowledge and improving generalization robustness.

3.3.3 Joint Training Strategy. To simultaneously optimize for accurate ranking of the ground-truth items and alignment with the ensemble consensus, we train the framework in a multi-task learning manner. The final objective function combines the recommendation loss derived from multi-source hard negatives (Equation 9) and the consensus distillation loss:

$$\mathcal{L} = \mathcal{L}_{rec} + \mu \cdot \mathcal{L}_{dis}, \quad (13)$$

where μ is a weighting coefficient that controls the strength of the distillation signal. By jointly optimizing this objective, our framework ensures that the self-model M_{self} not only discriminates the positive item from hard negatives effectively but also maintains a holistic understanding of the global preference distribution.

3.4 Time Complexity Analysis

We analyze the time complexity of the proposed framework across its three distinct stages. In the *multi-source scoring* stage, constructing the candidate pool requires $O(N)$, while dual-view scoring and teacher signal aggregation incur $O(Nd)$ and $O(N)$, respectively. Consequently, the complexity of this stage is bounded by $O(Nd)$. In the *divergence re-ranking* stage, computing divergence-aware scores operates in $O(N)$. The hard negative sampling procedure introduces a term of $O(N \log M)$, where M denotes the size of the random sampling pool. Since the optimization objective cost $O(d)$ is negligible compared to the sampling cost (i.e., $d \ll N \log M$), this

Table 2: Dataset statistics.

Dataset	Health	Sports	Beauty	Toys	LastFM	KuaiRand
# Users	66,519	35,598	22,363	19,412	1,090	18,642
# Items	28,237	18,357	12,101	11,924	3,646	7,566
# Inter	485,163	296,337	198,502	167,597	52,551	231,471
# AvgLen	7.3	8.3	8.8	8.6	48.2	12.4
Sparsity	99.97%	99.95%	99.93%	99.93%	98.68%	99.83%

stage scales as $O(N \log M)$. Finally, the *consensus distillation* stage involves probabilistic soft labeling and KL distillation, both of which operate in $O(N)$. Comparing the costs across stages, and given that $\log M \ll d$, the $O(N \log M)$ term from the re-ranking stage is dominated by the $O(Nd)$ term from the scoring stage. Therefore, the time complexity of the entire framework is $O(Nd)$.

This is on par with representative state-of-the-art baselines such as DNS+ and MixGCF, whose complexities are also $O(Nd)$. Therefore, our method does not incur additional asymptotic computational overhead, while fully utilizing the available computational cost to deliver improved performance.

4 Experiments

In this work, we aim to answer the following research questions:

- (RQ1) How does MDCNS perform compared to existing negative sampling methods in sequential recommendation?
- (RQ2) How well does MDCNS generalize across recommendation models, loss functions and weak to strong setting?
- (RQ3) How do components of MDCNS affect the performance?
- (RQ4) How does the loss of MDCNS evolve during training?

4.1 Experiments Setup

4.1.1 Datasets. Experiments are conducted on six real-world datasets: **Sports, Beauty, Toys and Health** are derived from Amazon product reviews [21], corresponding to the "Sports and Outdoors", "Beauty", "Toys and Games" and "Health and Personal Care" categories, respectively. **KuaiRand** [8] is retrieved from the recommendation logs of the video-sharing mobile application Kuaishou. It notably contains millions of user interactions with randomly exposed items. **LastFM**² is a music artist recommendation dataset derived from user tagging behaviors. In this work, the user-assigned tags associated with artists are utilized as item attributes. For consistency with prior work [16, 48], we process the Amazon datasets by retaining only the 5-core filtered version where each item/user appears in at least five interactions. The KuaiRand and LastFM undergo identical preprocessing. Table 2 presents a comprehensive summary of the key statistics for our six processed datasets.

4.1.2 Evaluation Proposal. In alignment with established protocols [16, 48], we assess model performance utilizing Recall@K and NDCG@K across $K \in \{5, 10, 20\}$. Regarding dataset partitioning, we implement a ratio splitting scheme, allocating 70% of sequences for training, 20% for validation, and the remaining 10% for testing.

4.1.3 Backbone Models. The method proposed herein is inherently model-agnostic, enabling seamless integration into the training processes of a wide range of sequential recommendation architectures.

²<https://grouplens.org/datasets/hetrec-2011/>

Table 3: Performance comparison across six datasets. The best and the second-best scores are marked in bold and underlined fonts. RC and NG denote Recall and NDCG. * denotes p-value < 0.05 for paired t-tests.

Method	Beauty						Toys					
	RC@5	RC@10	RC@20	NG@5	NG@10	NG@20	RC@5	RC@10	RC@20	NG@5	NG@10	NG@20
RNS	0.0434	0.0612	0.0966	0.0289	0.0347	0.0435	0.0592	0.0865	0.1231	0.0406	0.0496	0.0587
DNS	0.0398	0.0545	0.0715	0.0288	0.0335	0.0378	0.0216	0.0288	0.0366	0.0170	0.0193	0.0213
MixGCF	0.0349	0.0550	0.0805	0.0232	0.0297	0.0361	0.0144	0.0175	0.0237	0.0106	0.0117	0.0132
AdaSIR	<u>0.0590</u>	0.0840	0.1127	<u>0.0417</u>	<u>0.0497</u>	<u>0.0568</u>	0.0685	0.0947	0.1241	0.0487	0.0573	0.0647
GNNO	<u>0.0577</u>	<u>0.0854</u>	0.1144	<u>0.0388</u>	<u>0.0476</u>	<u>0.0550</u>	<u>0.0705</u>	<u>0.0989</u>	<u>0.1292</u>	<u>0.0475</u>	<u>0.0566</u>	<u>0.0641</u>
DNS+	0.0550	0.0769	0.1140	0.0355	0.0425	0.0519	0.0721	0.0963	<u>0.1344</u>	0.0503	0.0581	<u>0.0676</u>
MixGCF+	0.0532	0.0720	0.1068	0.0359	0.0418	0.0506	0.0685	0.0968	0.1282	0.0463	0.0554	0.0635
SRNS	0.0487	0.0742	<u>0.1180</u>	0.0334	0.0416	0.0527	<u>0.0772</u>	0.0978	0.1308	<u>0.0521</u>	<u>0.0586</u>	0.0668
MDCNS	0.0823	0.1144	0.1502	0.0582	0.0685	0.0775	0.0896	0.1200	0.1447	0.0668	0.0765	0.0828
improve	39.49%*	33.95%*	27.29%*	39.56%*	37.83%*	36.44%*	16.06%*	21.33%*	7.66%*	28.21%*	30.55%*	22.49%*

Method	Sports						Health					
	RC@5	RC@10	RC@20	NG@5	NG@10	NG@20	RC@5	RC@10	RC@20	NG@5	NG@10	NG@20
RNS	0.0233	0.0368	0.0587	0.0153	0.0195	0.0250	0.0360	0.0533	0.0777	0.0239	0.0295	0.0357
DNS	0.0048	0.0087	0.0157	0.0027	0.0039	0.0056	0.0243	0.0311	0.0388	0.0196	0.0217	0.0236
MixGCF	0.0037	0.0059	0.0132	0.0023	0.0030	0.0048	0.0207	0.0274	0.0344	0.0165	0.0186	0.0204
AdaSIR	0.0250	0.0354	0.0522	0.0168	0.0202	0.0245	0.0404	0.0603	0.0893	0.0275	0.0340	0.0413
GNNO	<u>0.0326</u>	<u>0.0480</u>	<u>0.0697</u>	<u>0.0206</u>	<u>0.0255</u>	<u>0.0310</u>	<u>0.0458</u>	<u>0.0653</u>	<u>0.0955</u>	<u>0.0309</u>	<u>0.0372</u>	<u>0.0448</u>
DNS+	0.0312	0.0486	<u>0.0713</u>	0.0202	0.0258	0.0316	0.0453	<u>0.0681</u>	<u>0.0968</u>	0.0316	0.0389	<u>0.0461</u>
MixGCF+	0.0323	<u>0.0489</u>	0.0683	<u>0.0215</u>	<u>0.0269</u>	<u>0.0318</u>	<u>0.0492</u>	0.0653	0.0950	<u>0.0334</u>	0.0386	0.0461
SRNS	0.0315	0.0478	0.0680	0.0212	0.0265	0.0316	0.0469	0.0668	0.0922	0.0328	<u>0.0393</u>	0.0456
MDCNS	0.0402	0.0573	0.0806	0.0285	0.0340	0.0398	0.0619	0.0857	0.1147	0.0439	0.0516	0.0588
improve	23.31%*	17.18%*	13.04%*	32.46%*	26.39%*	25.18%*	25.81%*	25.84%*	18.49%*	31.44%*	31.29%*	27.54%*

Method	LastFM						KuaiRand					
	RC@5	RC@10	RC@20	NG@5	NG@10	NG@20	RC@5	RC@10	RC@20	NG@5	NG@10	NG@20
RNS	0.0917	0.1483	0.2244	0.0591	0.0773	0.0966	0.0349	0.0606	0.111	0.0214	0.0298	0.0424
DNS	0.0782	0.1209	0.1678	0.0515	0.0652	0.0769	0.0349	0.0638	0.0976	0.0216	0.0310	0.0395
MixGCF	0.0735	0.1156	0.1675	0.0461	0.0597	0.0729	0.0295	0.0413	0.0622	0.0216	0.0255	0.0307
AdaSIR	0.1061	0.1728	0.2552	0.0691	0.0905	0.1114	0.0354	0.0606	0.1115	0.0218	0.0299	0.0428
GNNO	<u>0.1185</u>	<u>0.1879</u>	<u>0.2692</u>	<u>0.0790</u>	<u>0.1013</u>	<u>0.1218</u>	<u>0.0548</u>	<u>0.0847</u>	<u>0.1426</u>	<u>0.0334</u>	<u>0.0447</u>	<u>0.0591</u>
DNS+	0.1154	0.1768	0.2569	0.0745	0.0943	0.1145	0.0542	<u>0.0923</u>	0.1416	<u>0.0344</u>	<u>0.0459</u>	<u>0.0592</u>
MixGCF+	0.1121	0.1682	0.2474	0.0694	0.0873	0.1073	0.0547	0.0847	0.1340	0.0326	0.0423	0.0547
SRNS	0.1106	0.1739	0.2505	0.0728	0.0932	0.1126	0.0477	0.0804	0.1239	0.0321	0.0425	0.0534
MDCNS	0.1363	0.2033	0.2869	0.0915	0.1131	0.1342	0.0617	0.0992	0.1592	0.0368	0.0487	0.0637
improve	15.02%*	8.20%*	6.58%*	15.82%*	11.65%*	10.18%*	12.59%*	7.48%*	11.64%*	6.98%*	6.10%*	7.60%*

To evaluate its generalization ability and effectiveness, we conduct a series of extensive experiments using several representative backbone models from diverse architectural paradigms.

- **MLP-based: FMLP4Rec** [49] represents an all-MLP architecture that encodes sequential dependencies via learnable filters.
- **RNN-based: GRU4Rec** [9] uses Gated Recurrent Units to model the temporal evolution of user interests effectively.
- **Transformer-based: SASRec** [16] leverages multi-head self-attention to capture long-range dependencies within sequences.
- **Linear Transformer-based: LinRec** [20] adopts a linearized attention mechanism for efficient sequence modeling.
- **State Space Model-based: Mamba4Rec** [19] applies the Selective State Space Model to achieve efficient sequence processing.

4.1.4 Baselines. We compare MDCNS with various state-of-the-art hard negative-focused methods and false negative mitigated methods on sequential recommendation models:

- **RNS** [25] employs a uniform distribution to select negatives.
- **Hard negative-focused methods:**
 - (1) **DNS** [44] is a dynamic hard negative sampling technique, where negatives are chosen based on the highest item scores.
 - (2) **MixGCF** [15] generates hard negatives by mixing positive signals from the interaction graph into negative samples through both positive and hop mixing strategies.
 - (3) **AdaSIR** [1] employs a two-stage process that maintains a sample pool, with importance resampling to select negatives.
- **False negative-mitigated methods:**

Table 4: Performance of different sequential recommendation models on different real-world datasets using BCE loss [28]. The best scores are marked in bold fonts. RC and NG denote Recall and NDCG. * denotes p-value < 0.05 for paired t-tests.

Domain	Method	SASRec _{BCE}				LinRec _{BCE}				FMLP4Rec _{BCE}			
		RC@5	RC@10	NG@5	NG@10	RC@5	RC@10	NG@5	NG@10	RC@5	RC@10	NG@5	NG@10
Beauty	RNS	0.0434	0.0612	0.0289	0.0347	0.0523	0.0751	0.0363	0.0435	0.0532	0.0800	0.0342	0.0429
	DNS+	0.0550	0.0769	0.0355	0.0425	0.0617	0.0939	0.0420	0.0525	0.0635	0.0903	0.0432	0.0519
	MDCNS	0.0823*	0.1144*	0.0582*	0.0685*	0.0818*	0.1162*	0.0579*	0.0689*	0.0872*	0.1198*	0.0621*	0.0726*
Toys	RNS	0.0592	0.0865	0.0406	0.0496	0.0639	0.0860	0.0431	0.0505	0.0592	0.0870	0.0416	0.0505
	DNS+	0.0721	0.0963	0.0503	0.0581	0.0742	0.1050	0.0521	0.0622	0.0808	0.1056	0.0522	0.0600
	MDCNS	0.0896*	0.1200*	0.0668*	0.0765*	0.0963*	0.1241*	0.0688*	0.0777*	0.0968*	0.1282*	0.0684*	0.0785*
Sports	RNS	0.0233	0.0368	0.0153	0.0195	0.0261	0.0433	0.0159	0.0215	0.0261	0.0416	0.0163	0.0213
	DNS+	0.0312	0.0486	0.0202	0.0258	0.0323	0.0472	0.0220	0.0268	0.0365	0.0494	0.0236	0.0278
	MDCNS	0.0402*	0.0573*	0.0285*	0.0340*	0.0475*	0.0618*	0.0313*	0.0359*	0.0447*	0.0635*	0.0302*	0.0363*

Table 5: Performance of the other three sequential recommendation models on two real-world datasets using BPR loss [25]. The best scores are marked in bold fonts. RC and NG denote Recall and NDCG. * denotes p-value < 0.05 for paired t-tests.

Domain	Method	SASRec _{BPR}				GRU4Rec _{BPR}				Mamba4Rec _{BPR}			
		RC@5	RC@10	NG@5	NG@10	RC@5	RC@10	NG@5	NG@10	RC@5	RC@10	NG@5	NG@10
Beauty	RNS	0.0416	0.0657	0.0300	0.0377	0.0241	0.0398	0.0154	0.0204	0.0527	0.0791	0.0343	0.0428
	DNS+	0.0523	0.0760	0.0351	0.0427	0.0340	0.0501	0.0231	0.0283	0.0572	0.0760	0.0351	0.0427
	MDCNS	0.0791*	0.1109*	0.0569*	0.0671*	0.0612*	0.0921*	0.0425*	0.0524*	0.0755*	0.1122*	0.0548*	0.0667*
Toys	RNS	0.0561	0.0865	0.0383	0.0481	0.0237	0.0371	0.0162	0.0205	0.0577	0.0860	0.0411	0.0501
	DNS+	0.0633	0.0953	0.0445	0.0546	0.0299	0.0469	0.0170	0.0225	0.0690	0.0947	0.0484	0.0568
	MDCNS	0.0911*	0.1153*	0.0658*	0.0737*	0.0659*	0.0901*	0.0444*	0.0522*	0.0932*	0.1267*	0.0656*	0.0764*
Sports	RNS	0.0270	0.0424	0.0171	0.0220	0.0129	0.0205	0.0082	0.0106	0.0278	0.0390	0.0197	0.0233
	DNS+	0.0284	0.0416	0.0204	0.0245	0.0171	0.0267	0.0106	0.0136	0.0320	0.0466	0.0215	0.0262
	MDCNS	0.0433*	0.0604*	0.0293*	0.0348*	0.0320*	0.0469*	0.0206*	0.0253*	0.0441*	0.0629*	0.0308*	0.0368*

- (1) **GNNO** [7] identifies negatives based on neighborhood overlap and mitigates false negatives by evaluating similarity.
- (2) **DNS+** [29] enhances DNS by randomly sampling the top- M candidates to reduce the occurrence of false negatives.
- (3) **MixGCF+** [15] enhances MixGCF by randomly sampling the top- M candidates to reduce the occurrence of false negatives.
- (4) **SRNS** [6] employs a score-based memory and variance-based sampling technique to ensure high-quality negative samples.

4.2 Implementation Details

We adopt the official RecBole implementations [45, 46] for all backbone models and follow the hyperparameter configurations recommended in the original papers whenever applicable. For negative sampling baselines, we use the released implementations and hyperparameter search spaces reported in their respective works to ensure a fair comparison. All models are optimized with Adam [17] (learning rate $lr = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$), using an embedding size of 64 and a batch size of 1024. We set $N = 100$, and tune M over the range $\{1, 2, 3, 4, 5\}$. The balance coefficients α , β , and γ are selected via grid search in the interval $[0.5, 3.0]$ with a step size of 0.5, while the temperature τ is chosen from the set $\{0.7, 1.0, 10.0\}$. The coefficient μ is fixed to 1. Early stopping on the validation set with a patience of 30 is used to ensure training convergence. **During evaluation, we adopt a full item ranking protocol and use only the**

self model M_{self} to generate recommendations. To mitigate random variance, we report the final results on the test set as the average over three independent runs.

4.3 Performance of MDCNS (RQ1)

The comprehensive performance evaluation contrasting MDCNS against diverse baseline negative sampling frameworks across six datasets is detailed in Table 3. **For the experimental configuration, the foundational SASRec is employed as the self model, accompanied by Mamba4Rec acting as the peer model.** Both architectures are optimized using the standard binary cross-entropy (BCE) loss [28]. Empirical findings indicate that the introduced MDCNS reliably achieves superior performance across all evaluated metrics and data domains. Most impressively, MDCNS registers significant enhancements over the most competitive baselines, peaking at a 27.29% improvement in Recall@20 and a 36.44% surge in NDCG@20 on the Beauty dataset. It is also particularly noteworthy that the framework consistently yields state-of-the-art results on non-Amazon datasets (namely, LastFM and KuaiRand), thereby demonstrating exceptional domain generalization and data robustness.

(Answer to RQ1) MDCNS systematically outperforms existing negative sampling methods across diverse datasets, achieving state-of-the-art performance with substantial metric enhancements.

Table 6: Ablation study on three Amazon datasets. ($\downarrow$ MDCNS) denotes the performance drop from the full MDCNS method.

Variant	Beauty				Toys				Sports			
	RC@5	RC@10	NG@5	NG@10	RC@5	RC@10	NG@5	NG@10	RC@5	RC@10	NG@5	NG@10
MDCNS	0.0823	0.1144	0.0582	0.0685	0.0896	0.1200	0.0668	0.0765	0.0402	0.0573	0.0285	0.0340
w/o CD	0.0697	0.0957	0.0491	0.0573	0.0860	0.1138	0.0600	0.0689	0.0368	0.0531	0.0256	0.0308
($\downarrow$ MDCNS)	15.31%	15.64%	16.35%	16.35%	4.02%	5.17%	10.18%	9.93%	8.46%	7.33%	10.18%	9.41%
w/o CD-DR	0.0599	0.0831	0.0434	0.0510	0.0819	0.1066	0.0585	0.0665	0.0343	0.0503	0.0226	0.0278
($\downarrow$ MDCNS)	27.22%	25.43%	27.36%	25.55%	8.59%	11.17%	12.43%	13.07%	14.68%	12.22%	20.70%	18.24%
w/o CD-DR-MS	0.0550	0.0769	0.0355	0.0425	0.0721	0.0963	0.0503	0.0581	0.0312	0.0486	0.0202	0.0258
($\downarrow$ MDCNS)	33.17%	39.00%	32.78%	37.96%	19.53%	19.75%	24.70%	24.05%	22.39%	15.18%	29.12%	24.12%

Table 7: Performance of MDCNS under weak to strong setting.

Domain	Method	RC@5	RC@10	NG@5	NG@10
Beauty	RNS	0.0434	0.0612	0.0289	0.0347
	DNS+	0.0550	0.0769	0.0355	0.0425
	MDCNS _{wts}	<u>0.0760</u>	<u>0.1140</u>	<u>0.0542</u>	<u>0.0663</u>
	MDCNS	0.0823	0.1144	0.0582	0.0685
Toys	RNS	0.0592	0.0865	0.0406	0.0496
	DNS+	0.0721	0.0963	0.0503	0.0581
	MDCNS _{wts}	0.0942	<u>0.1195</u>	<u>0.0650</u>	<u>0.0731</u>
	MDCNS	<u>0.0896</u>	0.1200	0.0668	0.0765
Sports	RNS	0.0233	0.0368	0.0153	0.0195
	DNS+	0.0312	0.0486	0.0202	0.0258
	MDCNS _{wts}	0.0433	0.0596	0.0299	0.0351
	MDCNS	<u>0.0402</u>	<u>0.0573</u>	<u>0.0285</u>	<u>0.0340</u>
LastFM	RNS	0.0917	0.1483	0.0591	0.0773
	DNS+	0.1154	0.1768	0.0745	0.0943
	MDCNS _{wts}	0.1384	0.2078	0.0921	0.1144
	MDCNS	<u>0.1363</u>	<u>0.2033</u>	<u>0.0915</u>	<u>0.1131</u>

4.4 Generalization Evaluation (RQ2)

In this section, we demonstrate the generalization abilities of MDCNS through experiments on different recommendation backbone models, various loss functions and weak to strong setting.

4.4.1 Generalization across Different Backbone Models. The proposed MDCNS is designed as a model-agnostic negative sampling framework, offering high flexibility across various sequential recommendation backbones. To validate this plug-and-play capability, the framework is deployed across a diverse suite of five backbone models, following the details in Section 4.1.3. In these configurations, Mamba4Rec generally serves as the peer model, with the exception of the Mamba4Rec backbone evaluation, where SASRec is utilized in this role. As evidenced by the results in Tables 4 and 5, MDCNS achieves significant and statistically robust performance gains across all tested backbones, consistently outperforming established baselines. Notably, MDCNS demonstrates distinct advantages across a spectrum of architectures, ranging from filter MLP models (e.g., FMLP4Rec) to advanced mamba models (e.g., Mamba4Rec). The substantial improvements observed even on relatively simpler RNN based GRU4Rec further underscore its exceptional robustness.

These compelling results substantiate that MDCNS can be universally and reliably applied to enhance the performance of a wide range of sequential recommendation systems.

4.4.2 Generalization Across Varied Loss Functions. In addition to its architectural flexibility, MDCNS demonstrates strong performance across a range of optimization objectives. To explore this versatility, we evaluate MDCNS using two widely-used loss functions: Binary Cross-Entropy (BCE) loss [28], which treats recommendation as a classification problem, and Bayesian Personalized Ranking (BPR) loss [25], which focuses on optimizing pairwise rankings. As detailed in Tables 4 and 5, MDCNS consistently outperforms baseline sampling methods across various datasets and evaluation metrics under both loss functions. This highlights that the performance gains with MDCNS are largely independent of the chosen optimization objective. These results affirm the robustness and flexibility of MDCNS across different learning paradigms, underlining its ability to generalize effectively and adapt to a wide range of recommendation tasks. This positions MDCNS as a highly versatile tool, well-suited for sequential recommendation across diverse settings.

4.4.3 Generalization across Weak to Strong Setting. To further assess the robustness of our framework in a "weak-to-strong" generalization paradigm, we examine the influence of a weaker peer model on overall performance. Specifically, we use SASRec as the backbone and Mamba4Rec as the peer model, denoted as MDCNS, and compare it with MDCNS_{wts}, where SASRec is the backbone and a weak model GRU4Rec serves as the peer model. As shown in Table 7, the performance dynamics follow a dualistic pattern. On the Beauty and Toys datasets, the standard model MDCNS outperforms MDCNS_{wts}. However, on the Sports and LastFM datasets, MDCNS_{wts} achieves competitive results, even marginally outperforming MDCNS. Importantly, both variants consistently show substantial improvements over the baselines, regardless of these dataset-specific variations. This highlights the structural robustness of our framework, demonstrating that its effectiveness is not strictly dependent on utilizing a state-of-the-art peer model. We attribute this resilience to our negative sampling mechanism, which disrupts the negative self-reinforcement loop and introduces essential diversity and difficulty by divergence re-ranking, even when signals are derived from a weaker model.

(Answer to RQ2) MDCNS exhibits exceptional generalization, consistently delivering robust performance gains across diverse backbones, varied loss functions, and weak-to-strong settings.

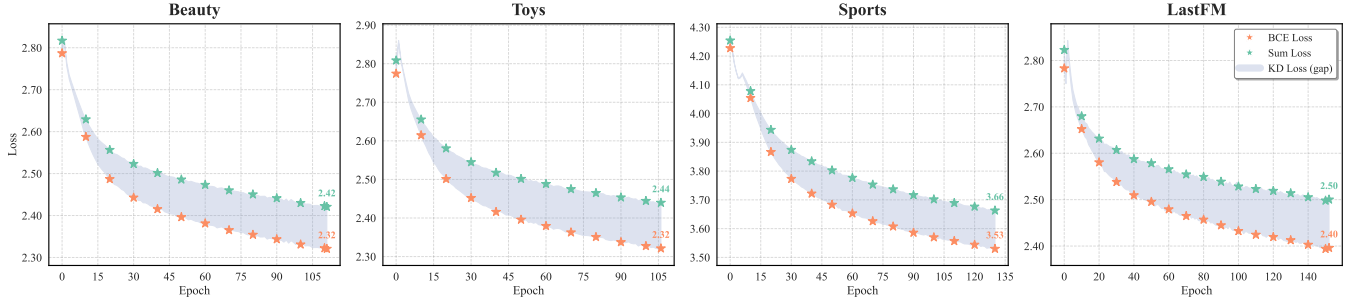


Figure 3: Loss dynamics of MDCNS on four datasets.

4.5 Ablation Study (RQ3)

In this section, we perform a series of ablation studies to examine the contribution of each component in MDCNS, namely multi-source scoring (MS), divergence re-ranking (DR), and consensus distillation (CD). All experiments are conducted on the SASRec backbone model, using three Amazon datasets. The results are in Table 6. The following variants are evaluated in our study:

- **w/o CD:** This variant excludes consensus distillation and directly utilizes the recommendation loss, $\mathcal{L}_{rec}$, for model training.
- **w/o CD-DR:** In this variant, divergence re-ranking is removed, meaning the score differences between M_{self} and M_{peer} in the candidate set are not used. Instead, hard negative samples are selected using the original scores.
- **w/o CD-DR-MS:** This variant omits multi-source negative scoring, relying exclusively on the scores from M_{self} to mine hard negative samples, without incorporating negative samples from other sources, which degenerates to the base DNS+ method.

As shown in Table 6, MDCNS consistently outperforms all ablation variants. Removing the consensus distillation phase ("w/o CD") results in a clear performance drop, highlighting the importance of extracting information from the teacher's consensus. Further exclusion of divergence re-ranking ("w/o CD-DR") leads to noticeable degradation, confirming its effectiveness in selecting more diverse negative samples. The additional decline observed in the absence of multi-source sampling ("w/o CD-DR-MS") further underscores the crucial role of multi-source negative sampling in breaking the negative self-reinforcement loop.

(Answer to RQ3) The progressive performance degradation across all ablation variants confirms that multi-source scoring, divergence re-ranking, and consensus distillation are all mutually indispensable for the optimal performance of MDCNS.

4.6 Training Dynamics Analysis

To empirically validate the stability and convergence of our optimization process, we analyze the training loss trajectories visualized in Figure 3. We observe that both the BCE recommendation loss and the cumulative sum loss exhibit a steady, monotonic decrement, eventually stabilizing at a minimal bound. This convergence behavior attests to the robustness of our proposed framework. A pivotal observation lies in the evolution of the knowledge distillation loss, represented by the gap between the curves. Unlike conventional distillation where the gap often diminishes rapidly, our KD

loss maintains a significant magnitude or expands throughout the training phase. We attribute this phenomenon to the proposed divergence re-ranking mechanism. By actively preventing the teacher and self-models from collapsing into homogenized representations, this mechanism ensures that the teacher continuously provides an informative gradient for the student. This sustained knowledge transfer capability further corroborates the effectiveness of our approach from the perspective of optimization dynamics.

(Answer to RQ4) MDCNS exhibits robust training dynamics where primary losses converge steadily, yet distillation loss expands to provide sustained informative gradients for optimization.

5 Related Works

In this section, we present a literature review regarding sequential recommendation and negative sampling.

5.1 Sequential Recommendation

Sequential Recommendation (SR) aims to capture the dynamic evolution of user preferences to predict future engagements. The research landscape has witnessed a paradigm shift from early probabilistic transition models [26] and matrix factorization (MF) techniques [11] to sophisticated neural architectures. In the early deep learning era, GRU4Rec [9, 10] pioneered the use of recurrent units, while convolutional architectures such as Caser [31] and NextIt-Net [42] were introduced to capture local temporal patterns. This focus subsequently shifted toward self-attention mechanisms, with SASRec [16] and Bert4Rec [30] becoming the standards for modeling long-range dependencies. More recently, the frontier has been further extended by frequency-domain models like FilterMLP [49], as well as selective state-space architectures, exemplified by Mamba4Rec [19] and RecMamba [40]. Beyond architectural refinements, the community has increasingly embraced auxiliary learning paradigms. Notably, contrastive learning frameworks (e.g., CL4SRec and ICLRec [4, 24, 37]), causal inference [36], and distributionally robust optimization (DRO) [39, 50] have been explored as parallel paradigms to alleviate data sparsity and bolster model robustness in real-world recommendation scenarios.

5.2 Negative Sampling

Negative sampling is fundamentally essential for sequential recommendation models trained on implicit feedback to obtain valid contrastive supervision [38]. Early methodologies primarily depended on heuristic distributions, operationalized through uniform

strategies like Random Negative Sampling (RNS) [25] or frequency-biased approaches such as Popularity-based Negative Sampling (PNS) [3, 47]. The paradigm subsequently shifted toward dynamic mechanisms. Notably, Dynamic Negative Sampling (DNS) [44] pioneered this transition by prioritizing unobserved items with elevated predicted scores. This trajectory was further enriched by adversarial generative architectures [22, 33] and auxiliary-data-driven models, prominently featuring methods like SamWalker and MixGCF [2, 6, 15, 41]. Currently, the community is actively addressing the intrinsic bottlenecks of hard negative mining from two primary perspectives. On one hand, frameworks including DNS+ [29] and GNNO [7] are dedicated to eliminating "false negative" noise. On the other hand, recent advances introduced by EXHANS and R²NS [18, 35] reveal that the premature introduction of excessively hard negatives often triggers early-stage training collapse. Consequently, curriculum learning strategies [34] have been increasingly adopted to dynamically pace the sampling difficulty throughout the training process. However, existing methods still face three key issues: negative self-reinforcement loop, restricted sampling diversity and inefficient resource allocation. Distinct from existing studies, we introduce a novel sampling method that disrupts the negative self-reinforcement loop via multi-source scoring, enhances sample diversity through divergence re-ranking, and capitalizes on computational resources via consensus distillation.

6 Conclusion

In this paper, we presented MDCNS, a novel negative sampling framework inspired by ZPD theory that effectively addresses the negative self-reinforcement loop, restricted sampling diversity, and inefficient resource allocation. MDCNS integrates three key components: multi-source sampling, which breaks the vicious cycle of self-selection by leveraging peer and teacher signals; divergence re-ranking, which enhances sampling diversity by quantifying the prediction discrepancy between views; and consensus distillation, which optimizes training efficiency by aligning the model with the teacher's global consensus. Extensive experiments across diverse datasets, backbones and loss functions demonstrate that MDCNS consistently outperforms state-of-the-art methods, achieving superior performance and generalization. While we fully utilized the computational cost, the multi-source architecture inevitably introduces additional overhead during the training phase. Future work will dedicate to exploring lighter and more efficient alternatives.

Acknowledgements

This work is supported in part by National Natural Science Foundation of China (No. 62422215 and No. 62472427), Major Innovation & Planning Interdisciplinary Platform for the "DoubleFirst Class" Initiative, Renmin University of China, Public Computing Cloud, Renmin University of China, fund for building world-class universities (disciplines) of Renmin University of China, the Outstanding Innovative Talents Cultivation Funded Programs 2024 of Renmin University of China.

References

- [1] Jin Chen, Defu Lian, Binbin Jin, Kai Zheng, and Enhong Chen. 2022. Learning recommenders for implicit feedback with importance resampling. In *Proceedings of the ACM Web Conference 2022*. 1997–2005.
- [2] Jiawei Chen, Can Wang, Sheng Zhou, Qihao Shi, Yan Feng, and Chun Chen. 2019. Samwalker: Social recommendation with informative sampling strategy. In *The world wide web conference*. 228–239.
- [3] Ting Chen, Yizhou Sun, Yue Shi, and Liangjie Hong. 2017. On sampling strategies for neural network-based collaborative filtering. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 767–776.
- [4] Yongjun Chen, Zhiwei Liu, Jia Li, Julian McAuley, and Caiming Xiong. 2022. Intent Contrastive Learning for Sequential Recommendation. *arXiv preprint arXiv:2202.02519* (2022).
- [5] Jingtao Ding, Yuhuan Quan, Xiangnan He, Yong Li, and Depeng Jin. 2019. Reinforced negative sampling for recommendation with exposure data.. In *IJCAI Macao*, 2230–2236.
- [6] Jingtao Ding, Yuhuan Quan, Quanming Yao, Yong Li, and Depeng Jin. 2020. Simplify and robustify negative sampling for implicit collaborative filtering. *Advances in Neural Information Processing Systems* 33 (2020), 1094–1105.
- [7] Lu Fan, Jiahu Pu, Rongsheng Zhang, and Xiao-Ming Wu. 2023. Neighborhood-based hard negative mining for sequential recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2042–2046.
- [8] Chongming Gao, Shijun Li, Yuan Zhang, Jiawei Chen, Biao Li, Wenqiang Lei, Peng Jiang, and Xiangnan He. 2022. Kuairand: An unbiased sequential recommendation dataset with randomly exposed videos. In *Proceedings of the 31st ACM international conference on information & knowledge management*. 3953–3957.
- [9] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).
- [10] Balázs Hidasi, Massimo Quadroni, Alexandros Karatzoglou, and Domonkos Tikk. 2016. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 241–248.
- [11] Balázs Hidasi and Domonkos Tikk. 2016. General factorization framework for context-aware recommendations. *Data Mining and Knowledge Discovery* 30, 2 (2016), 342–371.
- [12] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).
- [13] Yujing Hu, Qing Da, Anxiang Zeng, Yang Yu, and Yinghui Xu. 2018. Reinforcement learning to rank in e-commerce search engine: Formalization, analysis, and application. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 368–377.
- [14] Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. 2020. Embedding-based retrieval in facebook search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2553–2561.
- [15] Tinglin Huang, Yuxiao Dong, Ming Ding, Zhen Yang, Wenzheng Feng, Xinyu Wang, and Jie Tang. 2021. Mixgcf: An improved training method for graph neural network-based recommender systems. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*. 665–674.
- [16] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 197–206.
- [17] Diederik P Kingma. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [18] Yuanzi Li, Xuri Ge, Jingyu Zhao, Yidan Wang, Jiyuan Yang, Zhumin Chen, Zhaochun Ren, and Xin Xin. 2026. R2NS: Recall and Re-ranking of Negative Samples for Sequential Recommendation. In *Proceedings of the ACM Web Conference 2026* (United Arab Emirates) (WWW '26). Association for Computing Machinery, New York, NY, USA, 6331–6341. <https://doi.org/10.1145/3774904.3792360>
- [19] Chengkai Liu, Jianghao Lin, Jianling Wang, Hanzhou Liu, and James Caverlee. 2024. Mamba4rec: Towards efficient sequential recommendation with selective state space models. *arXiv preprint arXiv:2403.03900* (2024).
- [20] Langming Liu, Liu Cai, Chi Zhang, Xiangyu Zhao, Jingtao Gao, Wanyu Wang, Yifu Lv, Wenqi Fan, Yiqi Wang, Ming He, et al. 2023. Linrec: Linear attention mechanism for long-term sequential recommender systems. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 289–299.
- [21] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. 43–52.
- [22] Dae Hoon Park and Yi Chang. 2019. Adversarial sampling and training for semi-supervised information retrieval. In *The World Wide Web Conference*. 1443–1453.
- [23] Tao Qi, Fangzhao Wu, Chuhuan Wu, Yongfeng Huang, and Xing Xie. 2020. Privacy-preserving news recommendation model learning. *arXiv preprint arXiv:2003.09592* (2020).
- [24] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. 2022. Contrastive learning for representation degeneration problem in sequential recommendation. In *Proceedings of the fifteenth ACM international conference on web search and data mining*. 813–823.
- [25] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).
- [26] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*. 811–820.
- [27] Karim Shabani, Mohamad Khatib, and Saman Ebadi. 2010. Vygotsky's zone of proximal development: Instructional implications and teachers' professional development. *English language teaching* 3, 4 (2010), 237–248.
- [28] Claude E Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal* 27, 3 (1948), 379–423.
- [29] Wentao Shi, Jiawei Chen, Fuli Feng, Jizhi Zhang, Junkang Wu, Chongming Gao, and Xiangnan He. 2023. On the theories behind hard negative sampling for recommendation. In *Proceedings of the ACM Web Conference 2023*. 812–822.
- [30] 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. *arXiv:1904.06690* [cs.IR] <https://arxiv.org/abs/1904.06690>
- [31] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (2018). <https://api.semanticscholar.org/CorpusID:39847715>
- [32] Tim Van Erven and Peter Harremoës. 2014. Rényi divergence and Kullback-Leibler divergence. *IEEE Transactions on Information Theory* 60, 7 (2014), 3797–3820.
- [33] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. 2017. Irgan: A minimax game for unifying generative and discriminative information retrieval models. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. 515–524.
- [34] Xin Wang, Yudong Chen, and Wenwu Zhu. 2021. A Survey on Curriculum Learning. *arXiv:2010.13166* [cs.LG] <https://arxiv.org/abs/2010.13166>
- [35] Yidan Wang, Xuri Ge, Xin Chen, Ruobing Xie, Su Yan, Xu Zhang, Zhumin Chen, Jun Ma, and Xin Xin. 2025. Exploration and Exploitation of Hard Negative Samples for Cross-Domain Sequential Recommendation. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining* (Hanover, Germany) (WSDM '25). Association for Computing Machinery, New York, NY, USA, 669–677. <https://doi.org/10.1145/3701551.3703535>
- [36] Zhenlei Wang, Shiqi Shen, Zhipeng Wang, Bo Chen, Xu Chen, and Ji-Rong Wen. 2022. Unbiased sequential recommendation with latent confounders. In *Proceedings of the ACM web conference 2022*. 2195–2204.
- [37] Xu Xie, Fei Sun, Zhao Yang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive learning for sequential recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*. IEEE, 1259–1273.
- [38] Lanling Xu, Jianxun Lian, Wayne Xin Zhao, Ming Gong, Linjun Shou, Daxin Jiang, Xing Xie, and Ji-Rong Wen. 2022. Negative sampling for contrastive representation learning: A review. *arXiv preprint arXiv:2206.00212* (2022).
- [39] Jiyuan Yang, Yue Ding, Yidan Wang, Pengjie Ren, Zhumin Chen, Fei Cai, Jun Ma, Rui Zhang, Zhaochun Ren, and Xin Xin. 2024. Debiasing Sequential Recommenders through Distributionally Robust Optimization over System Exposure. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*. 882–890.
- [40] Jiyuan Yang, Yuanzi Li, Jingyu Zhao, Hanbing Wang, Muyang Ma, Jun Ma, Zhaochun Ren, Mengqi Zhang, Xin Xin, Zhumin Chen, et al. 2024. Uncovering Selective State Space Model's Capabilities in Lifelong Sequential Recommendation. *arXiv preprint arXiv:2403.16371* (2024).
- [41] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 974–983.
- [42] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M Jose, and Xiangnan He. 2019. A simple convolutional generative network for next item recommendation. In *Proceedings of the twelfth ACM international conference on web search and data mining*. 582–590.
- [43] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM computing surveys (CSUR)* 52, 1 (2019), 1–38.
- [44] Weinan Zhang, Tianqi Chen, Jun Wang, and Yong Yu. 2013. 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*. 785–788.
- [45] Wayne Xin Zhao, Yupeng Hou, Xingyu Pan, Chen Yang, Zeyu Zhang, Zihan Lin, Jingsen Zhang, Shuqing Bian, Jiakai Tang, Wenqi Sun, et al. 2022. Recbole 2.0: (2020).

- Towards a more up-to-date recommendation library. In *Proceedings of the 31st ACM international conference on information & knowledge management*. 4722–4726.
- [46] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, et al. 2021. Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In *proceedings of the 30th acm international conference on information & knowledge management*. 4653–4664.
- [47] Jin Peng Zhou, Ga Wu, Zheda Mai, and Scott Sanner. 2020. Noise contrastive estimation for autoencoding-based one-class collaborative filtering. *arXiv preprint arXiv:2008.01246* (2020).
- [48] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM international conference on information & knowledge management*. 1893–1902.
- [49] Kun Zhou, Hui Yu, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Filter-enhanced MLP is all you need for sequential recommendation. In *Proceedings of the ACM web conference 2022*. 2388–2399.
- [50] Rui Zhou, Xian Wu, Zhaopeng Qiu, Yefeng Zheng, and Xu Chen. 2023. Distributionally robust sequential recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 279–288.