Article

Not peer-reviewed version

A Survey on Robust Sequential Recommendation: Fundamentals, Challenges, Taxonomy, and Future Directions

Yatong Sun , Xiaochun Yang , Bin Wang * , Yan Wang , Zhu Sun *

Posted Date: 28 January 2026

doi: 10.20944/preprints202601.2218.v1

Keywords: sequential recommender systems; robust sequential recommenders; unreliable instances

 doi:10.20944/preprints202601.2218.v1

Article

A Survey on Robust Sequential Recommendation: Fundamentals, Challenges, Taxonomy, and Future Directions

Yatong Sun [1], Xiaochun Yang [1], Bin Wang [2,*], Yan Wang [3] and Zhu Sun [4,*]

[1] Software College, Northeastern University
[2] School of Computer Science and Engineering, Northeastern University, China; National Frontiers Science Center for Industrial Intelligence and Systems Optimization, China; Key Laboratory of Data Analytics and Optimization for Smart Industry (Northeastern University), Ministry of Education
[3] School of Computing, Macquarie University
[4] Information Systems Technology and Design, Singapore University of Technology and Design
* Correspondence: binwang@mail.neu.edu.cn (B.W.); zhu_sun@sutd.edu.sg (Z.S.)

Abstract

In the era of information overload, sequential recommender systems (SRSs) have become indispensable tools for modeling users' dynamic preferences, assisting personalized decision-making and information filtering, and thus attracting significant research and industrial attention. Conventional SRSs operate on a critical assumption that every input interaction sequence is reliably matched with the target subsequent interaction. However, this assumption is frequently violated in practice: real-world user behaviors are often driven by extrinsic motivations—such as behavioral randomness, contextual influences, and malicious attacks—which introduce perturbations into interaction sequences. These perturbations result in mismatched input-target pairs, termed *unreliable instances*, which corrupt sequential patterns, mislead model training and inference, and ultimately degrade recommendation accuracy. To mitigate these issues, the study of **R**obust **S**equential **R**ecommenders (**RSRs**) has emerged as a focal point. This survey provides the first systematic review of advances in RSR research. We begin with a thorough analysis of unreliable instances, detailing their causes, manifestations, and adverse impacts. We then delineate the unique challenges of RSRs, which are absent in non-sequential settings and general denoising tasks. Subsequently, we present a holistic taxonomy of RSR methodologies and a systematic comparative analysis based on eight key properties, critically assessing the strengths and limitations of existing approaches. We also summarize standard evaluation metrics and benchmarks. Finally, we identify open issues and discuss promising future research directions. To support the community, we maintain a rich repository of RSR resources at https://github.com/AlchemistYT/Awesome-RSRs.

Keywords: sequential recommender systems; robust sequential recommenders; unreliable instances

1. Introduction

In the era of information saturation, recommender systems (RSs) have become indispensable pillars for personalized decision-making across digital platforms like e-commerce, entertainment, and social networking [1,2]. They create a win-win scenario by enhancing user experiences through filtering irrelevant content, boosting item exposure for merchants, and driving engagement and revenue for platforms. Among diverse RS paradigms, sequential recommender systems (SRSs) [1,3,4] stand out for their unique capacity to capture users' dynamic, evolving preferences—an ability absent in non-sequential approaches (content-based [5–7] or collaborative filtering [8–10]).

The Importance of Robust Sequential Recommendation. The fundamental assumption underlying SRSs—that every input interaction sequence reliably predicts the subsequent action as the target—proves untenable in practice: (i) Theoretical works in human-computer interaction [11] and

 doi:10.20944/preprints202601.2218.v1

psychology [12] demonstrate that user behaviors are driven not only by intrinsic preferences but also by *extrinsic motivations*—including behavioral randomness (e.g., accidental clicks), contextual influences (e.g., social contagion), and malicious manipulations (e.g., fake engagements). These extrinsic motivations introduce perturbations into interaction sequences, resulting in mismatched input-target pairs termed *unreliable instances* [13–15]. (ii) Empirical studies substantiate the pervasiveness of unreliable instances: 5% $\sim$ 13% of instances in popular SRS benchmarks are unreliable due to behavioral randomness and contextual influences [13,14], while malicious activity exacerbates this issue–TikTok identified 10.47 billion fake likes in Q4 2024 [16], and Facebook reported 1.4 billion counterfeit accounts in the same period [17]. Such unreliable instances corrupt sequential patterns, degrade recommendation accuracy, and harm all stakeholders in the recommendation ecosystem. To develop systems that maintain recommendation accuracy despite the presence of unreliable instances, the study of **R**obust **S**equential **R**ecommenders (**RSRs**) has emerged as a critical research frontier, evident in the publication trends shown in Figure 1. However, the field currently lacks a systematic review of these efforts. This survey aims to bridge this gap by providing the first comprehensive overview of RSR research.

Figure 1. The statistics of publications related to RSRs regarding the publication year and venue.

Differentiation from Existing Surveys. Despite the wealth of surveys in the RS literature, no work has yet offered a holistic investigation into robustness against unreliable instances for sequential recommendation. Existing relevant surveys can be categorized into three streams, each with critical limitations that our work addresses:

- **General SRSs** [1,3,4] provide extensive coverage of model architectures and learning paradigms. However, they operate under the assumption of clean, well-matched input-target pairs, overlooking the pervasive issue of data reliability in real-world user behavior sequences.
- **Robustness in RS** [2,18,19] discusses robustness broadly, focusing on security aspects like adversarial attacks and privacy. Yet, they neglect sequential dependencies and the specific manifestations of unreliability that are central to sequential recommendation.
- **Denoising Techniques** for computer vision (CV) and natural language processing (NLP) tasks [20,21] offer valuable methodologies for handling label noise. However, their application to sequential recommendation is problematic, as user interactions lack objective 'ground-truth' labels. Furthermore, these techniques are ill-equipped to handle the high-dimensional item spaces and the complex, sparse nature of recommendation data.

As summarized by Table 1, our survey bridges these critical gaps by presenting the first systematic review dedicated to RSR. We uniquely analyze the root causes and distinct manifestations of unreliable instances in the context of sequential recommendation (Section 3), delineate the unique challenges absent in other domains (Section 4), and provide a comprehensive taxonomy and comparative analysis of RSR methodologies tailored to these challenges (Section 5). By unifying these insights, our work establishes a foundational roadmap for this emerging field.

 doi:10.20944/preprints202601.2218.v1

Table 1. Comparison of our survey with related surveys. ○, △, and × indicate a property is fully satisfied, partially satisfied, and unsatisfied, respectively.

Survey Category	Primary Focus	Addresses Unreliable Instances?	Handles Sequential Data?	Specific to SRS Context?
General SRS [1,3,4]	Model architectures and learning paradigms for SRSs	×	○	○
Robustness in RS [2,18,19]	Robustness for non-sequential RSs	△	×	×
CV/NLP Denoising [20,21]	Label noise in CV and NLP tasks	△	×	×
Our Survey (RSR)	Robustness against unreliable instances in sequential recommendation	○	○	○

Survey Methodology. To ensure a high-quality survey, we adopted a systematic literature review methodology. We identified pertinent articles by querying major computer science repositories (e.g., Scopus and DBLP) and supplemented with Google Scholar to capture works not indexed in primary databases. Our comprehensive review covers RSR literature from 2015 to 2025, spanning top-tier conferences and journals such as NeurIPS, ICML, ICLR, SIGKDD, TheWebConf, SIGIR, ICDE, AAAI, IJCAI, WSDM, CIKM, RecSys, TPAMI, TKDE, TOIS, and TNNLS. The search utilized keywords including 'robust', 'robustness', 'noise', 'denoising', 'sequential recommendation', 'session recommendation', 'preference shift', 'preference drift' and their combinations. Our search focuses on five axes: (1) characteristics of unreliable instances, (2) unique challenges faced by RSRs, (3) technical innovations for robustness, (4) strengths and limitations of existing RSRs, and (5) evaluation metrics and benchmarks. We summarize a curated repository of RSR resources at https://github.com/AlchemistYT/Awesome-RSRs.

Main Contributions of This Survey. This survey establishes the first comprehensive review for RSRs. We unify fragmented research into a cohesive framework, offering the following key advancements. (1) *Systematic Analysis of Unreliable Instances*: We formalize the RSR problem by characterizing causes, manifestations, and impacts of unreliable instances, and their relationship to noisy data, laying the foundation of the survey. (2) *Identification of Unique Challenges*: We delineate unique challenges in RSR distinct from non-sequential recommendation and denoising tasks in CV/NLP. (3) *Comprehensive Taxonomy*: We propose a lifecycle-oriented taxonomy categorizing RSR methods based on robustness integration stages. (4) *Comparative Analysis Framework*: We systematically assess RSRs across eight key properties, elucidating methodological strengths and limitations. (5) *Standardized Evaluation Protocol*: We consolidate metrics and benchmarks for evaluating RSRs. (6) *Future Research Roadmap*: We identify critical open problems and emerging opportunities for advancing RSRs. (7) *Curated Resource Repository*: We provide a public repository with curated papers, code, and datasets in the field of RSRs.

Structure of the Survey. The remainder of this survey is organized as follows: Section 2 establishes the background and preliminaries of RSRs. Section 3 presents a systematic analysis of unreliable instances, detailing their causes, manifestations, and adverse impacts. Section 4 highlights the unique challenges inherent to RSRs. Section 5 introduces a comprehensive taxonomy and a systematic comparison for RSRs. Section 6 consolidates standard evaluation metrics and benchmarks for RSRs. Section 7 discusses open issues and promising future research directions. Finally, Section 8 provides a succinct conclusion to this comprehensive survey.

2. Background and Fundamentals

This section establishes the foundational concepts for RSR, including (i) the construction of instances, (ii) the training and inference paradigms of SRSs, and (iii) the definition of 'robustness' for SRSs.

2.1. The Construction of Data Instances for Sequential Recommendation

Let $\mathcal{U}$ and $\mathcal{V}$ denote the sets of users and items, respectively. Each user $u \in \mathcal{U}$ chronologically interacts with a sequence of items, denoted as $S^u = [v_1^u, v_2^u, ..., v_{|S^u|}^u]$, where $v_t^u \in \mathcal{V}$ is the t-th item that user u interacts with, and $|S^u|$ indicates the length of sequence S^u. To model users' dynamic

 doi:10.20944/preprints202601.2218.v1

preferences, conventional SRSs typically predict a target item v_t^u given its fixed-length sequence of preceding items $\mathbf{x}_t^u = [v_{t-L}^u, ..., v_{t-2}^u, v_{t-1}^u]$ as input, where L is the predefined input length. Such an input-target pair $\langle \mathbf{x}_t^u, v_t^u \rangle$ constitutes a data instance for SRSs.

To construct such instances, a common practice is to apply a sliding window of length $L+1$ over each user's interaction sequence as illustrated in Figure 2 [13–15,22]. In each step of sliding, the last item in the window is treated as the target, and the preceding L items serve as the input. This process splits a sequence S^u into $|S^u| - 1$ instances, where the last one is allocated to the test set $\mathcal{D}_{test}$, while the remaining instances populate the training set $\mathcal{D}_{train}$.

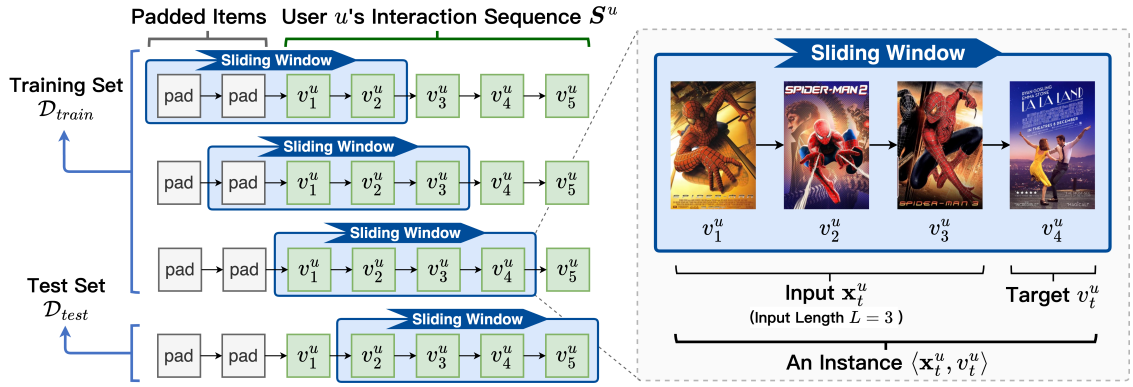

Figure 2. An example showing how an interaction sequence is split into data instances by a sliding window. In each step of sliding, the last item in the sliding window is treated as the target of an instance, while the preceding items in the window serve as the input.

2.2. The Training and Inference Paradigms of Conventional SRSs

Conventional SRSs are trained to maximize the matching score between the input and target of each instance, which can be formalized as minimizing the following objective function:

$$\mathcal{L}_{\text{srs}} = \sum_{\langle \mathbf{x}_t^u, v_t^u \rangle \in \mathcal{D}_{train}} \phi(f(\mathbf{x}_t^u, v_t^u)), \tag{1}$$

where f denotes the recommender that calculates the matching score between the input and target, ϕ is the recommendation loss function, which can be implemented by either pointwise loss (e.g. Cross Entropy loss [23]) that maximizes $f(\mathbf{x}_t^u, v_t^u)$, or pairwise loss (e.g. BPR loss [24]) that maximizes the gap between $f(\mathbf{x}_t^u, v_t)$ and scores for sampled negative targets.

During inference, the recommender f generates a Top-K recommendation list for a given input sequence $\mathbf{x}_t^u$ by ranking all items in the catalog $\mathcal{V}$:

$$R_t^u = \text{argsort}(\{f(\mathbf{x}_t^u, v) | v \in \mathcal{V}\})[0:K], \tag{2}$$

where argsort$(\cdot)$ returns items in descending order of their matching scores with the input.

2.3. The Definition of Robustness for Sequential Recommendation

Conventional SRSs operate under the assumption that each input sequence reliably predicts the subsequent interaction as the target. However, this assumption is frequently violated due to unreliable instances with mismatched input-target pairs, which are formally defined as follows:

Definition 1 (Unreliable Instance). *An instance $\langle \mathbf{x}_t^u, v_t^u \rangle$ is deemed unreliable if there exists an item $v \in \mathbf{x}_t^u$ that is mismatched with the target v_t^u.*

These unreliable instances corrupt sequential patterns and degrade recommendation accuracy by misleading both training (Equation 1) and inference (Equation 2) of SRSs. To address this, robust sequential recommenders (RSRs) are designed to maintain performance despite such instances. The core concept of 'robustness' in this context is formalized as ϵ-robust:

 doi:10.20944/preprints202601.2218.v1

Definition 2 (ϵ-robust). *Let ψ be an ideal corrector capable of transforming any unreliable instance into a perfectly matched input-target pair, formalized as $\langle \mathbf{x}_t^{u*}, v_t^{u*} \rangle = \psi(\mathbf{x}_t^u, v_t^u)$. Let f be an SRS trained on the original training set $\mathcal{D}_{train}$, and f^* be an oracle SRS trained on the corrected dataset $\mathcal{D}_{train}^*$, obtained by applying ψ to $\mathcal{D}_{train}$. For a tolerance $\epsilon > 0$, f is ϵ-robust if for every test instance $\langle \mathbf{x}_t^u, v_t^u \rangle \in \mathcal{D}_{test}$:*

$$\max_{v \in \mathcal{V}} |f(\mathbf{x}_t^u, v) - f^*(\psi(\mathbf{x}_t^{u*}, v))| \leq \epsilon. \tag{3}$$

Achieving ϵ-robust requires an RSR to approximate the oracle f^* that has access to purified data, entailing distinct criteria during training and inference phases as illustrated by Figure 3:

- **Training-phase Robustness**: During training, the RSR must precisely identify items within the input sequence that are genuinely relevant to the target (i.e., driven by the same intrinsic motivations). By focusing on these items, the model avoids learning erroneous patterns from perturbations [15,25].
- **Inference-phase Robustness**: During inference, the target is unobservable. The RSR must infer the underlying motivations from the input and ensure the recommendation list provides complete coverage for these motivations, without being skewed by perturbations [26,27].

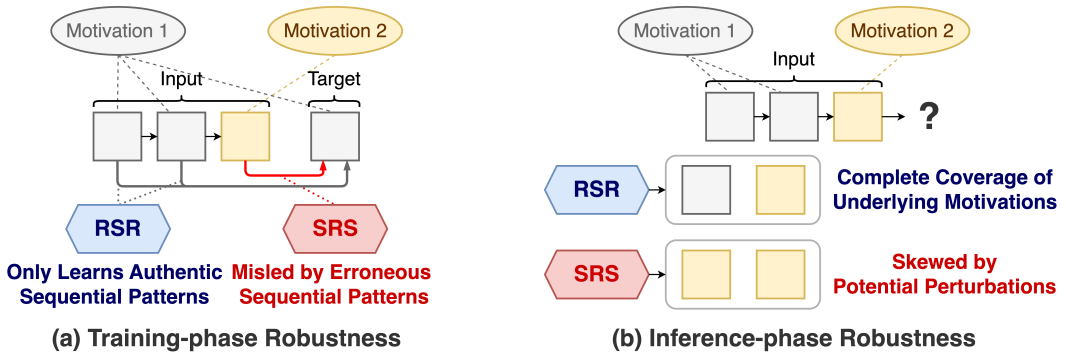

Figure 3. Training- and inference-phase robustness for sequential recommendation.

3. Unreliable Instances in Sequential Recommendation

The efficacy of SRSs hinges on the foundational assumption that input sequences reliably predict their subsequent actions. This section deconstructs the violation of this assumption by systematically analyzing unreliable instances: we elucidate their root causes, delineate their distinct manifestations, detail their adverse impacts on the recommendation ecosystem (summarized in Figure 4), and finally distinguish them from general data noise in both conceptual and practical dimensions.

 doi:10.20944/preprints202601.2218.v1

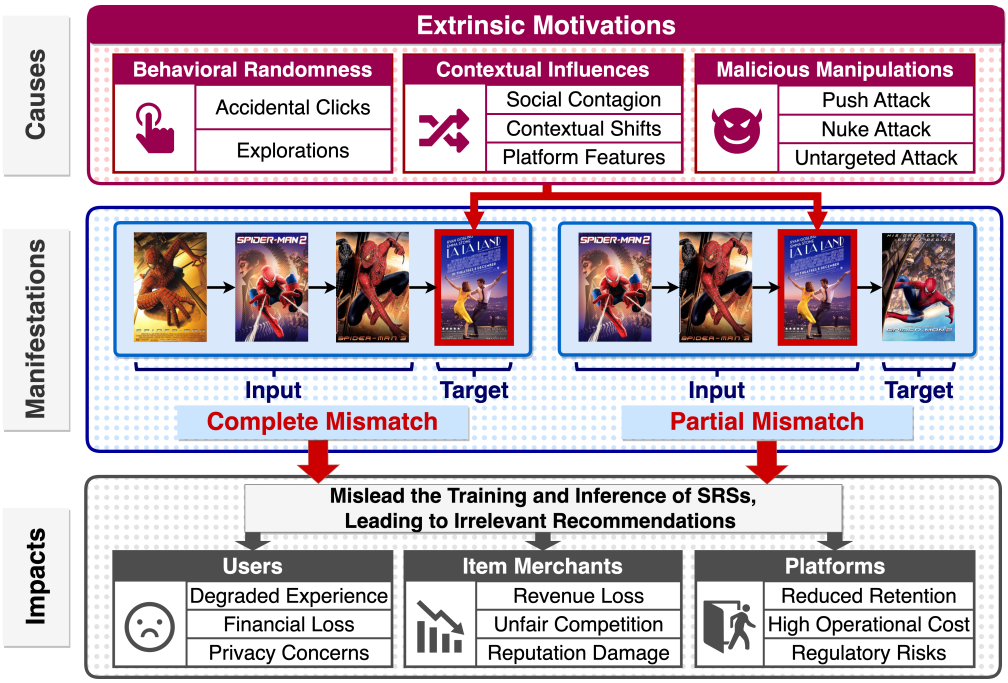

Figure 4. The causes, manifestations, and adverse impacts of unreliable training instances.

3.1. The Causes of Unreliable Instances

Theoretical studies in the domains of computer-human interaction [11] and psychology [12] reveal that real-world user interactions are not only determined by users' intrinsic preferences, but also by pervasive extrinsic motivations in practical scenarios. These extrinsic motivations—categorized as behavioral randomness, contextual influences, and malicious manipulations—introduce perturbations into users' interaction sequences, causing unreliable instances with mismatched input-target pairs. This subsection systematically analyzes each cause of unreliable instances, supported by empirical evidence and domain-specific examples.

Cause 1: Behavioral Randomness. User interactions often exhibit randomness that deviates from genuine preferences, posing significant challenges to preference modeling. Key examples include (1) *Accidental Interactions*: Misclicks or mis-taps introduce perturbations into interaction sequences. Empirical studies [28,29] show that 4.7%–6.3% of clicks are accidental, with a higher incidence on mobile interfaces than desktop platforms. (2) *Explorations*: Users may occasionally explore items outside their usual interests due to curiosity [30–32] or fatigue from monotonous recommendations [33]. Such explorations create transient patterns between irrelevant items, thereby misleading sequential preference modeling.

Cause 2: Contextual Influences. Contextual factors unrecorded in interaction logs systematically disrupt sequences. Typical forms include (1) *Social Contagion*: Interactions driven by peer pressure or influencer endorsements may misrepresent users' intrinsic preferences, introducing social biases into sequential patterns [34]. (2) *Contextual Shifts*: Temporal anomalies (e.g., holidays and festivals) [35], spatial variations (e.g., business trips) [36], and device-specific contexts (e.g., shared accounts) [37] may induce non-stationarity that disrupt sequence coherence. (3) *Platform Features*: Platform mechanisms like auto-play (e.g., YouTube's next-video autoplay) or discount promotion (e.g., simultaneous sales of irrelevant items) passively guide user actions, decoupling interactions from users' intrinsic preferences.

Cause 3: Malicious Manipulations. Malicious users in real-world scenarios deliberately inject perturbations into data to manipulate recommendation outcomes [19,38,39], with goals including (1) *Push Attack*: Inflate the exposure of specific items (e.g., merchant self-promotion). (2) *Nuke Attack*: Demote the exposure of specific items (e.g., merchants suppressing competitors' products). (3) *Untargeted Attack*: Undermining the platform's recommendation accuracy to erode user trust (e.g., attacks from competing platforms). Technically, existing manipulation approaches fall into two paradigms: (a)

 doi:10.20944/preprints202601.2218.v1

Model-agnostic Manipulations inject perturbations heuristically, without leveraging recommenders' internal architecture or training logic. While easy to implement, their effectiveness is limited by the lack of adaptation to model-specific characteristics. (b) *Model-aware Manipulations* craft perturbations optimized to maximize impact by exploiting recommenders' structure or training process. Such tailored manipulations severely compromise recommendation robustness.

3.2. The Manifestations of Unreliable Instances

As formalized in Section 2, each data instance in sequential recommendation comprises an *input* sequence $\mathbf{x}_t^u$ and a *target* item v_t^u. Perturbations caused by extrinsic motivations lead to mismatches between inputs and targets, which manifest in two distinct forms:

- *Complete Mismatch* [13,14]: This occurs when the target item v_t^u is itself a perturbation, rendering it completely mismatched with the input sequence. For example, as shown in Figure 4, a romantic film like 'La La Land' as the perturbed target is completely irrelevant to a preceding input sequence of superhero movies.
- *Partial Mismatch* [15,25]: This arises when the input sequence $\mathbf{x}_t^u$ contains one or more perturbed items, causing partial misalignment with the target. In Figure 4, when the romantic film 'La La Land' acts as a perturbed input item, it disrupts the coherence of a sequence targeting a superhero movie.

3.3. The Adverse Impacts of Unreliable Instances

Unreliable instances mislead the training and inference of SRSs, resulting in irrelevant recommendations and inflicting multi-faceted adverse impacts on users, item merchants, and platforms. This subsection elaborates on these impacts across the three stakeholder groups.

Adverse Impacts for Users: (1) *Degraded User Experience*. Unreliable instances distort sequential patterns, resulting in recommendations that misalign with user preferences (e.g., suggesting romantic films following superhero movies). This triggers user frustration, reduces satisfaction, and increases cognitive load due to manual filtering. (2) *Financial Loss*. SRSs misguided by unreliable instances may steer users toward purchases of irrelevant items, resulting in unnecessary expenditure. Worse still, malicious manipulations can promote fraudulent products, exposing users to financial losses from untrustworthy content. (3) *Privacy Concerns*. Artificially skewed recommendations from malicious attacks can erode trust in platform data integrity, raising user concerns about privacy violations and data misuse.

Adverse Impacts for Item Merchants: (1) *Revenue Erosion*. Misguided recommendations may prioritize irrelevant items to users, hindering merchants from reaching target audiences. This directly suppresses conversion rates and item sales. (2) *Unfair Competition.* Malicious merchants may inject perturbations to manipulate recommendations, promoting their own items or demoting competitors'. This practice distorts market fairness and drives out reputable players, degrading the overall item quality of the platform. (3) *Reputation Damage.* Persistent irrelevant recommendations may associate merchants' products with poor quality. For example, users tend to blame the merchant rather than the SRS for unsuitable purchases, harming brand reputation.

Adverse Impacts for Platforms: (1) *Reduced User and Merchant Retention.* Degraded recommendation accuracy diminishes user engagement and merchant revenue, potentially causing churn among both groups and undermining platform sustainability. (2) *Escalated Operational Costs*. Detecting and mitigating unreliable instances requires significant resources. For example, Facebook invests billions annually in security measures like fake account removal [17], diverting funds from technical innovation. (3) *Regulatory Risks*. Failure to address unreliable instances erodes public trust and invites legal penalties under regulations like the European Union's General Data Protection Regulation (GDPR) [40], which imposes fines for inadequate control of malicious activities (e.g., fake accounts or reviews).

 doi:10.20944/preprints202601.2218.v1

3.4. The Relationship Between Unreliable Instances and General Data Noise

Unreliable instances share conceptual ground with data noise, defined as instances with mislabeled targets in domains like CV and NLP [20]. Indeed, data noise represents a subset of unreliable instances, as mislabeled targets inherently create input-target mismatches, satisfying Definition 1. However, unreliable instances are not synonymous with data noise due to critical distinctions in causes, manifestations, and countermeasures:

(1) *Distinction in Causes*: Data noise typically stems from data corruption errors (e.g., mislabeled images or misspelled words) that undermine label correctness [20]. In sequential recommendation, however, user interactions lack objective 'correctness'. For example, watching a romantic film after superhero movies is not erroneous but may be prompted by extrinsic motivations irrelevant to genuine sequential patterns. Thus, unreliable instances arise from extrinsic motivations rather than annotation errors, precluding straightforward correctness judgments.

(2) *Distinction in Manifestations*. Data noise manifests solely as complete mismatch (due to erroneous targets), while unreliable instances include both complete and partial mismatches. In other words, data noise focuses exclusively on imperfections in targets, whereas unreliable instances encompass imperfections both in inputs as well as targets.

(3) *Distinction in Countermeasures*. Noise mitigation often relies on clean validation sets, expert annotations, or small-scale label transition matrices [20]. However, in sequential recommendation, three key barriers render such strategies infeasible. First, no dataset is entirely free of unreliable instances, eliminating the possibility of relying on 'clean' validation data. Second, obtaining subjective annotations for instance reliability is impractical: accurately inferring the motivations behind individual user behaviors is challenging, even for domain experts. Psychological studies show that even users themselves may be uncertain about the motivations behind their past behaviors [41]. Third, the large size of item catalogs (e.g., billions of items on platforms like Amazon or YouTube) makes learning label transition matrices computationally intractable.

4. Unique Challenges Faced by RSRs

Addressing unreliable instances introduces a set of unique challenges, which are fundamentally distinct from those in non-sequential RSs and denoising tasks in CV and NLP. These challenges arise from the complex, dynamic, and often unobservable nature of user behavior sequences, as well as the absence of ground-truth reliability annotations. Building on the analysis of unreliable instances in Section 3, this section systematically delineates eight core challenges inherent to RSRs.

Challenge 1: The Lack of Explicit Annotations for Instance Reliability. A fundamental obstacle in RSRs is the absence of ground-truth annotations for instance reliability, particularly the motivations behind user interactions. Unlike CV or NLP tasks, where mislabeled data (e.g., incorrect image classifications) can be manually identified through clear criteria, sequential recommendation lacks objective benchmarks to assess interaction relevance. This ambiguity stems from the subjective nature of user behavior, which defies judgment via crowdsourcing, expert analysis, or even user self-reports, as psychological evidence indicates users may be unable to articulate their own historical motivations [41].

The lack of explicit reliability annotations impedes both RSR training and evaluation. During training, models cannot employ supervised learning to directly address unreliable instances, forcing reliance on imperfect proxies such as heuristic rules or self-supervised signals. For evaluation, common practices like injecting synthetic perturbations through random sequence manipulations [42] often fail to emulate the complexity of real-world unreliable instances, undermining the validity of robustness assessments.

Challenge 2: The Complexity of Users' Sequential Behavioral Patterns. Mitigating unreliable instances hinges on accurately discerning whether co-occurring items in an instance are genuinely relevant (i.e., matched). This is non-trivial due to the inherent complexity of users' sequential behavioral patterns, which give rise to two critical confounding phenomena. (i) *Rare Co-occurrence of*

 doi:10.20944/preprints202601.2218.v1

Relevant Items: items highly relevant to a user's preferences may co-occur infrequently in observed sequences—often due to low prevalence or limited exposure. Given RSRs' data-driven reliance on statistical patterns, these genuinely relevant but rarely co-occurring items risk being misclassified as mismatched. (ii) *Frequent Co-occurrence of Irrelevant Items*: items without genuine sequential relevance may co-occur frequently due to extraneous confounders, such as contextual influences (e.g., two items on sale simultaneously) or malicious manipulations (e.g., push attacks on specific items). This induced frequent co-occurrence in behavioral data makes RSRs prone to misclassifying irrelevant items as matched.

Consequently, RSRs must disentangle authentic preference signals from frequency-based correlations, requiring techniques that transcend simple co-occurrence metrics. This challenge is unique to sequential modeling, as non-sequential approaches ignore the temporal dependencies and contextual factors underlying these patterns.

Challenge 3: Divergent Causes of Unreliable Instances. As Section 3.1 details, unreliable instances stem from three divergent causes—behavioral randomness, contextual influences, and malicious manipulations—each requiring distinct mitigation strategies: (i) Behavioral randomness demands probabilistic modeling or uncertainty estimation [13,43] to quantify the likelihood of accidental interactions. (ii) Contextual influences require integrating auxiliary metadata [14] and causal inference [44] to estimate how interactions would unfold without contextual triggers. (iii) Malicious manipulations call for defenses like anomaly detection frameworks [45] or adversarial training [46], to identify and neutralize attacks.

Crucially, these causes are not mutually exclusive—they coexist in real-world scenarios—requiring a unified framework to address them concurrently. However, building such a framework faces two critical hurdles. (i) *Strategy Contradictions*: techniques effective against one cause may inadvertently exacerbate others. For example, leveraging metadata to account for unobserved external influences can expose new attack surfaces for malicious manipulations. (ii) *Operational Overhead*: dynamically routing instances to cause-specific mitigation subsystems introduces significant architectural complexity and inference latency. This challenge is unique to RSRs, as noise in domains like CV and NLP typically stems from a single source (e.g., erroneous crowdsourced labels).

Challenge 4: Divergent Manifestations of Unreliable Instances. As Section 3.2 outlines, unreliable instances in sequential recommendation exhibit two divergent manifestations: complete mismatch, where extrinsic motivations render the target item irrelevant to the input sequence, and partial mismatch, where the input sequence contains items irrelevant to the target. These manifestations necessitate divergent mitigation strategies: (i) Complete mismatch requires instance-level interventions (e.g., filtering or correction) to prevent unreliable targets from distorting model learning. (ii) Partial mismatch demands fine-grained, interaction-level repairs (e.g., item removal or weighting) to address perturbations within input sequences.

However, developing integrated solutions that address both mismatches is complicated by two key issues. (i) *Corrective Interference Risk*: mitigating one manifestation may induce the other. For example, correcting a complete mismatch via target replacement can introduce partial mismatch if the new target misaligns with prior items. (ii) *Precision-Preservation Trade-off*: approaches must balance corrective accuracy against data retention—overly aggressive interventions risk discarding genuine signals, while insufficient correction leaves residual perturbations. This manifestation divergence highlights the need for dynamic adaptation to mismatch types while preserving sequential coherence, a challenge largely absent in non-sequential and CV/NLP settings.

Challenge 5: High-Dimensional Reliability Estimation Space. In CV and NLP, a common robustness strategy involves learning a *label transition matrix*, which estimates the probability of a label being misassigned to another class [47–50]. While effective for tasks with compact label spaces (e.g., 10–100 classes in CV benchmarks), this approach is computationally intractable for RSRs due to three key constraints. (i) *Quadratic Complexity with Item Catalog Size*: the transition matrix $\mathbf{T} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ grows quadratically with item catalog size $|\mathcal{V}|$. Practical platforms like Amazon and YouTube typically operate

 doi:10.20944/preprints202601.2218.v1

with $|\mathcal{V}| \geq 10^8$ items [51,52], resulting in parameter spaces exceeding 10^{16} elements—prohibitively large for optimization. (ii) *Sparse Item Transitions*: realistic item transition matrices are highly sparse, as most item pairs never co-occur in practice. This sparsity renders standard estimation techniques statistically inefficient and numerically unstable. (iii) *Dynamic Item Catalog*: continuous item turnover (new items introduced, obsolete items retired) necessitates frequent matrix re-estimation, further escalating computational burdens. Consequently, full transition matrix-based methods are infeasible for RSRs, demanding alternative reliability estimation strategies for high-dimensional item spaces.

Challenge 6: Dilemma between Training- and Inference-Phase Robustness. As formalized in Section 2.3, RSRs face divergent robustness requirements across training and inference phases. Training-phase robustness demands precise selection of input items relevant to the target to prevent learning spurious patterns, while inference-phase robustness requires comprehensive coverage of users' potential motivations in recommendation lists.

This creates a precision-coverage dilemma: methods optimized for training (e.g., aggressive filtering of rarely co-occurring items) may be overly destructive during inference, removing items that reflect nuanced preferences. Conversely, approaches prioritizing inference-phase coverage may retain too many perturbations during training, corrupting learned patterns. This dual-phase challenge is largely absent in CV/NLP denoising, which focuses primarily on training data cleansing [53,54].

Challenge 7: Trilemma between Robustness, Personalization, and Scalability. Designing effective RSRs necessitates balancing three competing objectives: robustness (accurate unreliable instance mitigation), personalization (adaptation to individual preferences, including niche interests), and scalability (computational efficiency at web scale). These dimensions exhibit inherent trade-offs. (i) *Robustness-Personalization Conflict*: aggressive unreliability mitigation (e.g., strict filtering) often discards long-tail user-item interactions, exacerbating data sparsity and degrading recommendation quality for nuanced preferences. (ii) *Robustness-Scalability Tension*: fine-grained robustness mechanisms (e.g., transformer-based sequence correction [15,55]) incur superlinear complexity, making them computationally infeasible for billion-scale interaction logs.

This trilemma is exacerbated in sequential recommendation due to the exponential growth of sequence permutation space with history length, the reliance on multi-granular behavioral patterns for personalization, and real-time inference constraints. Thus, RSRs require adaptive, resource-aware frameworks that dynamically balance these objectives.

Challenge 8: Transformation between Intrinsic and Extrinsic Motivations. In real-world scenarios, an interaction initially driven by an extrinsic factor (e.g., accidental clicks) may evolve into intrinsic preferences as the user develops genuine interests in the item. Conversely, an intrinsic motivation (e.g. interest in cartoons) may gradually fade and transform into an extrinsic motivation.

This bidirectional transformation necessitates dynamic frameworks capable of continuously updating motivational attributions as preferences evolve. Static approaches risk two types of errors: prematurely discarding interactions that may become meaningful (losing valuable signals) or retaining outdated interactions that degrade robustness. Achieving an adaptive yet reliable balance remains an open research challenge.

5. Taxonomy and Comparative Analysis of RSRs

This section presents a systematic taxonomy and comparative analysis of existing RSRs, aiming to provide a structured overview of the methodological landscape and a critical assessment of their strengths and limitations. To establish a principled categorization, we anchor our taxonomy in the canonical lifecycle of SRSs, which sequentially encompasses: Model Design, Instance Construction, Model Training, and Model Inference (Figure 5). RSRs integrate robustness mechanisms at each of these stages to mitigate the adverse impacts of unreliable instances. Accordingly, we classify existing RSRs into four major paradigms:

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 28 January 2026 doi:10.20944/preprints202601.2218.v1

❶ **Architecture-centric RSRs** embed robustness directly into the model architecture through perturbation-resistant designs (e.g., gating mechanisms or diffusion models), ensuring stable internal representations despite perturbed sequences.

❷ **Data-centric RSRs** operate at the Instance Construction stage, focusing on cleansing training data before or during model training. They proactively identify and rectify mismatched input-target pairs (via selection, reweighting, or correction), thereby eliminating erroneous sequential patterns from the training process.

❸ **Learning-centric RSRs** introduce robustness during model training. Rather than modifying the data or core architecture, they leverage specialized training strategies (e.g., adversarial training, robust loss functions) to guide the model to learn genuine user preferences while diminishing the influence of unreliable instances.

❹ **Inference-centric RSRs** address robustness at the final model inference stage. Acknowledging that real-time input sequences may contain perturbations, these methods generate comprehensive and balanced recommendation lists that fully capture users' underlying motivations and avoid being skewed by perturbations.

Figure 5. The Taxonomy of RSRs.

 doi:10.20944/preprints202601.2218.v1

For each paradigm, we systematically review representative methods, elucidate their core ideas and methodological innovations, and conduct a critical comparative analysis leveraging a novel ***Assessment Framework***—proposed in this work by aligning with the unique challenges of RSRs (detailed in Section 4)—which encompasses eight key properties:

- **P1.** *Multi-cause Robustness*: Ability to address diverse extrinsic motivations (behavioral randomness, contextual influences, malicious manipulations) that induce unreliable instances.
- **P2.** *Dual-manifestation Robustness*: Capacity to handle both complete mismatch (perturbed targets) and partial mismatch (perturbed inputs).
- **P3.** *Dual-phase Robustness*: Capability to satisfy robustness requirements (Section 2.3) in both the training phase and the inference phase.
- **P4.** *Motivation Transformation Awareness*: Ability to model transformations between intrinsic and extrinsic motivations over time.
- **P5.** *Generality*: Compatibility with existing SRSs without extensive architectural modifications.
- **P6.** *Data Accessibility*: Independence from side information (e.g., item attributes, user demographics) beyond raw user-item interaction data.
- **P7.** *Scalability*: Efficiency in large-scale real-world scenarios.
- **P8.** *Theoretical Grounding*: Existence of formal theoretical guarantees (e.g., robustness bounds, convergence proofs) for the method's efficacy.

5.1. Architecture-Centric RSRs

Architecture-centric RSRs embed robustness directly into model architectures through perturbation-resistant designs, such as *Attention Mechanism*, *Memory Networks*, *Gating Networks*, *Graph Neural Networks*, *Time–frequency Transforms*, and *Diffusion Models*. These methods stabilize internal representations when processing perturbed sequences, thereby mitigating the impact of unreliable instances. Figure 6 illustrates the evolutionary trajectory of architecture-centric RSRs, while the following subsections detail the technical characteristics of each methodological group.

 doi:10.20944/preprints202601.2218.v1

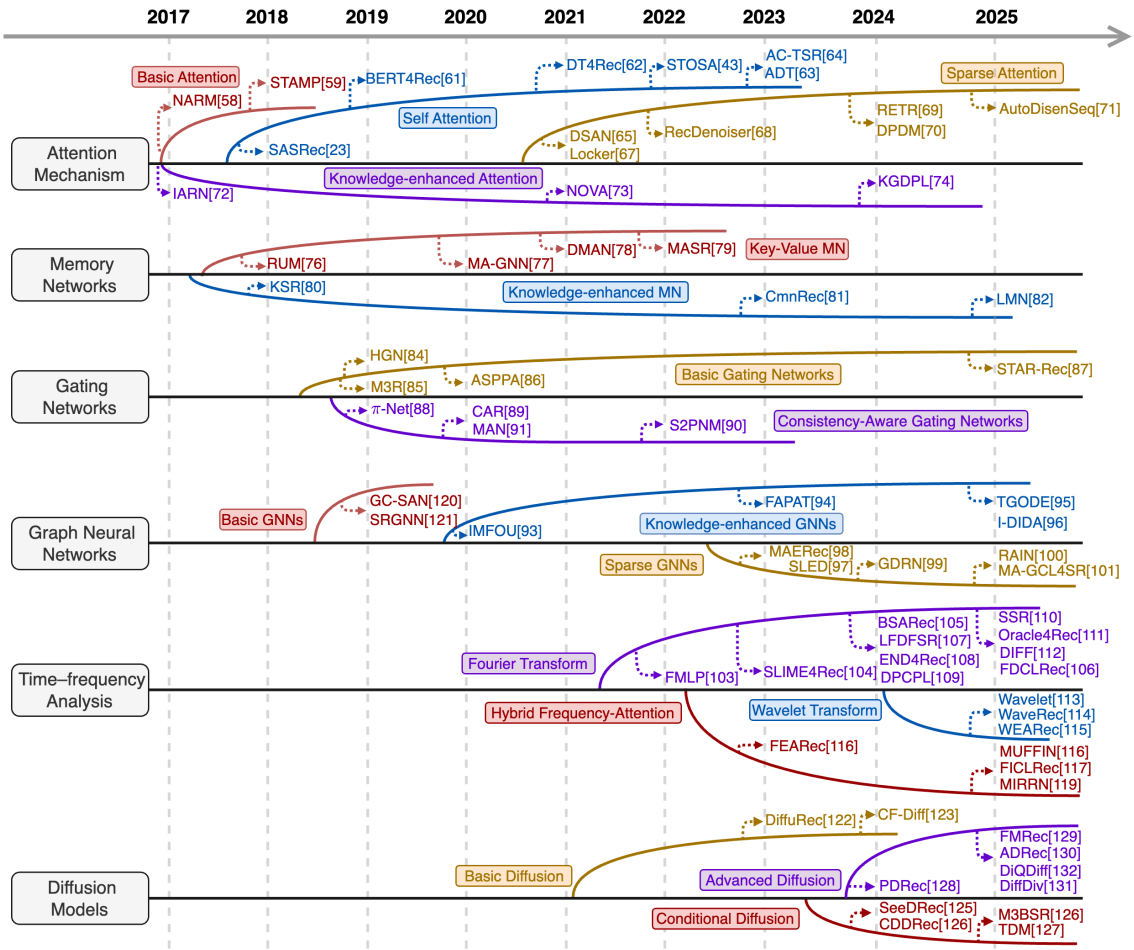

Figure 6. Development trajectory of Architecture-centric RSRs.

5.1.1. Architecture-Centric RSRs Based on Attention Mechanism

Attention mechanisms [56] enable dynamic weighting of input sequence components, naturally facilitating robustness by emphasizing relevant interactions and suppressing perturbations. We categorize attention-based RSRs into four subgroups: *Basic Attention*, *Self-Attention*, *Sparse Attention*, and *Knowledge-enhanced Attention*.

Subgroup 1: Basic Attention methods integrate attention modules into recurrent neural networks (RNNs) to dynamically weigh input items $v_i^u \in \mathbf{x}_t^u$:

$$a_i = \text{softmax}(\text{sim}(\mathbf{h}_i, \mathbf{q})) \tag{4}$$

where $\mathbf{h}_i$ denotes the hidden state of item v_i^u, $\mathbf{q}$ is a query vector (e.g., the embedding of the last interaction or session context), and $\text{sim}(\cdot,\cdot)$ is a similarity function (e.g., Cosine similarity [57]). **NARM** [58] combines RNN-based encoding with item-level attention to obtain sequence representation $\mathbf{c} = \sum_i a_i \mathbf{h}_i$, emphasizing relevant items while suppressing perturbations. **STAMP** [59] designs a short-term attention mechanism aligned with the last interaction, improving robustness to random behavioral fluctuations and contextual deviations.

Subgroup 2: Self Attention extends basic attention to multi-head self-attention [60], capturing salient commonality among input items as genuine preferences and eliminating uncommon signals as perturbations. For an input sequence embedding matrix $\mathbf{E}_{\text{seq}} \in \mathbb{R}^{L\times d}$, the self-attention output is:

$$\text{self-attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V} \tag{5}$$

where $\mathbf{Q} = \mathbf{E}_{\text{seq}}\mathbf{W}_1$, $\mathbf{K} = \mathbf{E}_{\text{seq}}\mathbf{W}_2$, and $\mathbf{V} = \mathbf{E}_{\text{seq}}\mathbf{W}_3$ are learnable projections. In particular, **SASRec** [23] uses left-to-right self-attention to adaptively attend to relevant historical items. **BERT4Rec** [61] employs bidirectional self-attention with a cloze task (masking random items and predicting them via sequence

 doi:10.20944/preprints202601.2218.v1

context) to form stable preference representations against perturbations. **DT4Rec** [62] and **STOSA** [43] model items as Gaussian distributions to capture behavioral uncertainty, using Wasserstein distance for similarity measurement to handle non-overlapping distributions and behavioral randomness. **ADT** [63] enhances Transformers with adaptive disentanglement (via independence and reconstruction objectives) to isolate perturbations by ensuring attention heads capture distinct preference aspects. **AC-TSR** [64] introduces spatial and adversarial calibrators to refine attention weights: the spatial calibrator incorporates low-level spatial features, while the adversarial calibrator corrects perturbations to highlight relevant items.

Subgroup 3: Sparse Attention approaches prune irrelevant items by inducing sparsity in attention weights. For example, **DSAN** [65] replaces softmax with a sparse transformation α-entmax:

$$\alpha\text{-entmax}(\mathbf{z}) = \text{ReLU}((1-\alpha)z_i + \alpha\tau)^{1/(1-\alpha)}, \tag{6}$$

where $\mathbf{z}$ is the pre-softmax attention score vector, α controls sparsity, τ ensures normalization, and $\text{ReLU}(\cdot)$ denotes the rectified linear unit function [66]. This transformation yields exact zeros in attention weights, effectively pruning perturbations. **Locker** [67] imposes local constraints on self-attention via masking-based encoders (deactivating distant tokens) to prioritize recent interactions and reduce the impact of irrelevant historical items. **RecDenoiser** [68] attaches differentiable binary masks to self-attention layers and optimizes them with unbiased gradient estimators. **RETR** [69] introduces pathway attention with Gumbel-Softmax sampling, enabling the model to focus on users' unique behavioral pathways while filtering trivial perturbations. **DPDM** [70] proposes a dual-perspective denoising model that achieves sparse attention via L_0-regularized graph reconstruction. **AutoDisenSeq** [71] automates attention design via neural architecture search, discovering optimal architectures for disentangling user intents and isolating signals from perturbations.

Subgroup 4: Knowledge-enhanced Attention methods integrate external knowledge to rectify attention scores, typically by fusing knowledge embeddings with sequence representations:

$$\mathbf{E}_{\text{enhanced}} = \text{fuse}(\mathbf{E}_{\text{seq}}, \mathbf{E}_{\text{kg}}), \tag{7}$$

where $\mathbf{E}_{\text{kg}}$ denotes knowledge-based embeddings for input items. Specifically, **IARN** [72] employs interacting attention gates that dynamically weight time steps in both user and item histories, with each user/item enriched by auxiliary information via a hierarchical feature encoder. **NOVA** [73] uses non-invasive self-attention (leveraging side information for query/key computation while keeping value projections based on pure item IDs) to refine attention distributions without information overwhelming. **KGDPL** [74] integrates knowledge graphs (KGs) into attention, using KG paths to distinguish genuine preferences from perturbations.

Discussion and Assessment (detailed in Table 2). Attention-based RSRs mitigate behavioral randomness via adaptive item weighting, effectively addressing partial mismatch. However, they lack defenses against contextual influences, malicious manipulations, neglect complete mismatch, motivational transformations, and inference-phase robustness. Practically, their non-plug-and-play design restricts generality; while most operate without auxiliary features, scalability is constrained by the quadratic complexity of self-attention. Moreover, they lack formal robustness guarantees.

5.1.2. Architecture-Centric RSRs Based on Memory Networks

Memory Network (MN) [75] enhances robustness via explicit, structured storage and retrieval of historical information, maintaining an external memory module to record users' interaction histories and enable selective reading/writing during sequence processing. By comparing current inputs with long-term behavioral representations, they can distinguish reliable patterns from perturbations. Existing memory network-based RSRs can be categorized into *Key–value MN* and *Knowledge-enhanced MN*.

Subgroup 1: Key-value MN employs key-value memory structures to store and retrieve user-specific information, maintaining a memory matrix $\mathbf{M}^u$ for each user u. The read operation computes the weight of each memory slot (each row vector of $\mathbf{M}^u$) as:

$$w_{t,k} = \text{softmax}(\beta \cdot \mathbf{v}_t^{u\top} \mathbf{m}_k^u), \tag{8}$$

 doi:10.20944/preprints202601.2218.v1

where $\mathbf{v}_t^u$ denotes the embedding of the target item, $\mathbf{m}_k^u$ is the k-th memory slot, and β is a scaling factor. The retrieved preference representation is $\mathbf{p}_u = \sum_{k=1}^{K} w_{t,k} \cdot \mathbf{m}_k^u$, which emphasizes input items that are relevant to the target and suppresses irrelevant ones. Write operations typically follow a first-in, first-out (FIFO) strategy to update memory with new interactions. Specifically, **RUM** [76] uses a user-specific memory matrix to store recent interactions, with an attention-based read operation to filter perturbations and FIFO writing to reduce outdated behavior impacts. **MA-GNN** [77] models short-term interests with a graph neural network and long-term interests with a key–value memory network, using a gating mechanism to adaptively combine them and mitigate perturbations. **DMAN** [78] employs a dynamic memory network with capsule-based routing to abstract long-term interests, smoothing perturbations via aggregation. **MASR** [79] employs balanced memory banks (centroid-wise and cluster-wise) to mitigate long-tail bias via retriever–copy mechanisms.

Subgroup 2: Knowledge-enhanced MN enriches memories with external knowledge to better discern perturbations, fusing knowledge-based information into item embeddings: $\tilde{\mathbf{v}}_i^u = \text{fuse}(\mathbf{v}_i^u, \mathbf{e}_{v_i^u})$, where $\mathbf{e}_{v_i^u}$ denotes the knowledge embedding for item v_i^u. The read operation explicitly considers relational information:

$$w_{t,k}^u = \text{softmax}\left(\beta \cdot \sum_{r}^{\mathcal{R}} \text{trans}(\tilde{\mathbf{v}}_t^u, r)^\top \text{trans}(\mathbf{m}_k^u, r)\right), \tag{9}$$

where $\mathcal{R}$ is the relation set, and $\text{trans}(\mathbf{v}, r)$ transforms embedding $\mathbf{v}$ to the subspace of relation r. Notably, **KSR** [80] uses a key–value memory network with KG relations as keys, focusing on attribute-level consistency to prune perturbations. **CmnRec** [81] introduces periodic/time-sensitive chunking with product quantization to accelerate memory networks, prioritizing recent, relevant signals and mitigating outdated perturbations. **LMN** [82] implements large-scale memory networks with cross features powered by NVLink optimization, enabling industrial deployment at the million-user scale.

Discussion and Assessment (detailed in Table 2). Memory-based RSRs address behavioral randomness via memory aggregation and partial mismatch through selective reading, but neglect contextual influences, malicious manipulations, and complete mismatch. They boost training-phase robustness via memory filtering yet overlook inference-phase robustness and fail to model motivational dynamics. Their user-specific storage hinders generality and scalability, and knowledge-enhanced variants require auxiliary features. Crucially, they lack formal robustness guarantees.

5.1.3. Architecture-Centric RSRs Based on Gating Networks

Gating networks [83] adaptively control information flow in neural architectures via learnable gates (binary/soft switches), determining which input items are relevant to the target and which are perturbations. Existing gating-based RSRs can be categorized into *Basic Gating Networks* and *Consistency-aware Gating Networks*.

Subgroup 1: Basic Gating Networks use multi-level/multi-scale gating to suppress perturbations at different granularities. Specifically, **HGN** [84] uses hierarchical gating (feature-level and item-level) to filter perturbations:

$$\mathbf{E}_{\text{feature}} = \mathbf{E}_{\text{seq}} \otimes \underbrace{\sigma(\mathbf{E}_{\text{seq}}\mathbf{W}_{g_1} + \mathbf{W}_{g_2}\mathbf{u})}_{\text{Feature-level Gating}}, \quad \mathbf{E}_{\text{item}} = \mathbf{E}_{\text{feature}} \otimes \underbrace{\sigma(\mathbf{E}_{\text{feature}}\mathbf{w}_{g_3} + \mathbf{W}_{g_4}\mathbf{u})}_{\text{Item-level Gating}} \tag{10}$$

where $\mathbf{E}_{\text{seq}} \in \mathbb{R}^{L \times d}$ denotes the sequence embedding matrix, $\mathbf{u} \in \mathbb{R}^d$ is the user embedding, $\otimes$ is element-wise multiplication, and $\sigma(\cdot)$ is the sigmoid function. The feature-level gating selects preference-aligned latent features, and the item-level gating downweights the importance of perturbation-prone items in sequences. **M3R** [85] dynamically combines predictions from tiny/short/long-range encoders via gating, emphasizing relevant temporal context to reduce perturbation impacts across scales. **ASPPA** [86] implements adaptive sequence partitioning via stacked RNNs with binary boundary detection gates, identifying semantic subsequences to filter context-induced perturbations. **STAR-Rec** [87] adopts preference-aware multi-head attention with gating to capture static item relationships and user-specific patterns, routing behavioral patterns to specialized experts via a mixture-of-experts prediction layer.

 doi:10.20944/preprints202601.2218.v1

Subgroup 2: Consistency-aware Gating Networks use consistency-driven gating to balance users' long- and short-term preferences: if short-term preferences deviate from long-term ones, the latter's importance should be suppressed. Notably, π-**Net** [88] uses shared account filter units with gating to disentangle mixed behavior sequences from multiple users. **CAR** [89] implements consistency-aware gating to balance general and current preference models, computing gate weights based on the divergence between recent items and historical sequence centroids. **S2PNM** [90] combines dictionary learning with gating to model dynamic preferences, using gates to control dictionary basis vector combinations and adapt to preference shifts. **MAN** [91] uses gating to balance predictions from parametric neural recommenders (capturing frequent behaviors) and non-parametric memory modules (capturing infrequent behaviors).

Discussion and Assessment (detailed in Table 2). Gating-based RSRs resist behavioral randomness via adaptive filtering, effectively addressing partial mismatch and boosting training-phase robustness. However, they overlook contextual influences, malicious manipulations, complete mismatch, motivational transformations, as well as inference-phase robustness. Practically, they are not plug-in modules. While basic gating is computationally efficient, hierarchical and multi-scale variants increase overhead, and formal robustness guarantees are absent.

5.1.4. Architecture-Centric RSRs Based on Graph Neural Networks

Graph Neural Networks (GNNs) [92] enhance robustness by modeling user–item interactions as graphs, capturing high-order dependencies and complex transition patterns. Neighborhood aggregation in GNNs smooths perturbations and highlights consistent behavioral signals, enabling the distillation of reliable patterns from perturbed sequences. Existing GNN-based RSRs can be classified into: *Basic GNNs*, *Knowledge-Enhanced GNNs*, *Sparse GNNs*, and *Dynamic GNNs*.

Subgroup 1: Basic GNNs construct directed graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ from user interaction sequences ($\mathcal{V}$: items; $\mathcal{E}$: consecutive interactions) and apply GNNs to capture item transitions. Node embeddings are updated as:

$$\mathbf{a}_i^l = \mathbf{A}_{i:}[\mathbf{v}_1^{l-1}, \ldots, \mathbf{v}_n^{l-1}]^\top \mathbf{W} + \mathbf{b}, \quad \mathbf{v}_i^l = \mathrm{GNN}(\mathbf{a}_i^l, \mathbf{v}_i^{l-1}), \tag{11}$$

where $\mathbf{A}$ is the adjacency matrix, $\mathbf{a}_i^l$ encodes the neighboring information of node v_i^u at layer l, and $\mathbf{v}_i^l$ is the layer-l embedding of v_i^u. Sequence representations combine local (last-click) and global (attentive) embeddings. Core works in this subgroup (**GCSAN** and **SRGNN**) combine GNNs with attention to capture local/global session dependencies, constructing dynamic session graphs and using graph-based attention to filter irrelevant items.

Subgroup 2: Knowledge-enhanced GNNs augment graph structures with external knowledge to improve reliability, fusing knowledge-based embeddings into item nodes: $\tilde{\mathbf{v}}_i = \mathrm{fuse}(\mathbf{v}_i, \mathbf{E}_{kg}(v_i))$, where $\mathbf{E}_{kg}(v_i)$ is the knowledge embedding of item v_i. This integration promotes attribute-level consistency and smooths perturbations in interaction graphs. In particular, **IMfOU** [93] models user intentions via attribute graphs (integrating ordered/unordered dependencies), filtering irrelevant interactions by emphasizing attribute-level patterns aligned with preferences. **FAPAT** [94] mines frequent attribute patterns from multiplex graphs to capture user intent, reducing perturbations from global item graphs by extracting compact attribute subgraphs. **TGODE** [95] constructs an item evolution graph to model uneven item distributions and irregular user interests, and integrates temporal guidance via a generalized graph neural ordinary differential equation to align the evolutionary processes of user preferences and item trends. **I-DIDA** [96] explicitly models the evolution of graph structures over time to account for dynamic user preferences and temporal perturbations, assuming that for a node and its temporal ego-graphs, predictive patterns can be decomposed into invariant and variant components (caused by perturbations).

Subgroup 3: Sparse GNNs prune unreliable graph connections to reduce perturbations, typically via adaptive edge masking:

$$\hat{\mathbf{A}}(i,j) = \begin{cases} \mathbf{A}(i,j), & \text{if } s_{i,j} > \Delta \\ 0, & \text{otherwise} \end{cases}, \tag{12}$$

 doi:10.20944/preprints202601.2218.v1

where $s_{i,j}$ is the similarity score between v_i and v_j, Δ is a pruning threshold, and $\hat{\mathbf{A}}$ is the sparsified adjacency matrix. Specifically, **SLED** [97] pre-trains a structure encoder to predict interaction reliability weights, denoising graphs by down-weighting unreliable instances based on structural patterns. **MAERec** [98] uses a graph masked autoencoder with adaptive path masking, reconstructing masked transitional paths and using task-adaptive loss to avoid noisy augmentations. **GDRN** [99] integrates graph diffusion with adaptive graph generation, sparsifying graphs via embedding similarity and using neural ordinary differential equations to reduce over-smoothing and noise. **RAIN** [100] performs graph/session-level denoising via self-supervised edge reconstruction, pruning noisy edges and re-weighting session items by contextual coherence. **MA-GCL4SR** [101] uses non-shared GNN encoders to generate diverse views and isolates relevant signals from noise via maximum mean discrepancy regularization.

Discussion and Assessment (detailed in Table 2). GNN-based RSRs resist behavioral randomness via neighborhood aggregation, effectively handling partial mismatch and enhancing training-phase robustness. Yet, they neglect contextual influences, malicious manipulations, complete mismatch, motivational transformations, and inference-phase robustness. Practically, they require architectural modification, with a few variants demanding auxiliary features. Scalability is constrained by costly full-graph computation at scale, and formal robustness guarantees are absent.

5.1.5. Architecture-Centric RSRs Based on Time–Frequency Analysis

Time–frequency analysis [102] transforms user interaction sequences into the frequency domain to attenuate perturbations and extract invariant patterns, leveraging spectral representations to isolate extrinsically motivated perturbations. Time–frequency-based RSRs can be categorized into: *Fourier Transform*, *Wavelet Transform*, and *Hybrid Frequency–attention*.

Subgroup 1: Fourier Transform employs Fast Fourier Transform (FFT) to project user interaction sequences into the frequency domain, where perturbations typically appear as high-frequency noise. Given a sequence embedding matrix $\mathbf{E}_{\text{seq}}$, the transformation and filtering of FFT is:

$$\mathbf{X} = \mathcal{F}(\mathbf{E}_{seq}), \quad \tilde{\mathbf{X}} = \mathbf{X} \otimes \mathbf{W}_{\text{f}}, \quad \tilde{\mathbf{E}}_{\text{seq}} = \mathcal{F}^{-1}(\tilde{\mathbf{X}}), \tag{13}$$

where $\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$ denote the FFT and inverse FFT, $\mathbf{W}_{\text{f}}$ is a learnable spectral filter, and $\tilde{\mathbf{E}}_{\text{seq}}$ is the denoised sequence embedding. Specifically, **FMLP** [103] replaces self-attention with an all-MLP architecture enhanced by learnable FFT filters for efficient denoising; **SLIME4Rec** [104] introduces dynamic frequency selection and static frequency split modules to adaptively suppress perturbations across temporal scales. **BSARec** [105] and **FDCLRec** [106] address self-attention's low-pass filtering limitation via frequency rescaling, amplifying high-frequency components to mitigate short-term pattern oversmoothing. **LFDFSR** [107] integrates FFT with side information fusion, using decoupled attention to handle perturbations and attribute heterogeneity. **END4Rec** [108] and **DPCPL** [109] combine Fourier-based denoising pre-training with prompt learning for parameter-efficient downstream adaptation while filtering perturbations. **SSR** [110] leverages spiking neural networks with FFT for on-device deployment, achieving energy-efficient perturbation handling via spike-based representations. **Oracle4Rec** [111] uses future information as oracle guidance, with FFT correcting historical sequence perturbations via forward-looking preference alignment. **DIFF** [112] implements dual side-information filtering and fusion, where frequency-domain processing separates attribute-level perturbations from genuine preference signals.

Subgroup 2: Wavelet Transform employs discrete wavelet transforms (DWT) instead of FFT to achieve multi-resolution analysis, capturing both time and frequency information to localize perturbations. The general formulation is:

$$\mathbf{W}_{\text{low}}, \mathbf{W}_{\text{high}} = \mathcal{W}(\mathbf{E}_{\text{seq}}), \quad \tilde{\mathbf{W}}_{\text{high}} = \beta^2 \otimes \mathbf{W}_{\text{high}}, \quad \tilde{\mathbf{E}}_{\text{seq}} = \mathcal{W}^{-1}(\mathbf{W}_{\text{low}}, \tilde{\mathbf{W}}_{\text{high}}), \tag{14}$$

where $\mathcal{W}$ and $\mathcal{W}^{-1}$ denote DWT and inverse DWT. The low-pass component $\mathbf{W}_{\text{low}}$ captures smooth, long-term stable behavior patterns (e.g., a user's persistent preference for comedy movies), reflecting intrinsic, noise-resistant preferences. The high-pass component $\mathbf{W}_{\text{high}}$ encodes fine-grained transient variations, encompassing both meaningful short-term dynamics and noisy patterns (e.g., accidental

 doi:10.20944/preprints202601.2218.v1

one-time interactions). A learnable attenuation factor β adaptively downweights noise in $\mathbf{W}_{\text{high}}$ while preserving meaningful details, enabling denoised sequence modeling. For example, **Wavelet** [113] integrates Mamba state-space models with wavelet neural filters, capturing non-stationary behaviors via time-frequency localization while maintaining linear complexity. **WaveRec** [114] and **WEARec** [115] demonstrate wavelet transforms' superiority over Fourier methods for non-periodic sequences, using diverse wavelet bases (Haar and Daubechies) to discern non-stationary signals and short-term fluctuations.

Subgroup 3: Hybrid Frequency-attention combines frequency-domain filtering with attention mechanisms, using the denoised frequency representation $\tilde{\mathbf{E}}_{\text{seq}}$ to compute self-attention projections in Equation 5. This hybrid design captures both global frequency patterns and local sequential dependencies. In particular, **FEARec** [116] introduces frequency ramp structures and auto-correlation mechanisms to capture multi-scale patterns and address self-attention's low-pass filtering bias. **MUFFIN** [117] implements user-adaptive frequency filtering via global/local modules, generating personalized attention across filters based on individual frequency characteristics. **FICLRec** [118] combines frequency redistribution with intent self-attention, using high-frequency and cluster-level center alignment losses to enhance perturbation discrimination. **MIRRN** [119] uses a multi-head Fourier transformer with target-aware attention for multi-granularity interest retrieval.

Discussion and Assessment (detailed in Table 2). Time–frequency-based RSRs mitigate behavioral randomness via spectral denoising, effectively addressing partial mismatch and boosting training-phase robustness. However, they lack resistance to contextual influences, malicious manipulations, complete mismatch, motivational transformations, and inference-phase robustness. Their reliance on FFT/wavelet operations requires architectural adaptation, with a few incorporating side information. While FFT and wavelet operations scale efficiently as $O(L \log L)$, hybrid attention modules increase computational cost, and formal robustness guarantees remain unexplored.

 doi:10.20944/preprints202601.2218.v1

Table 2. Evaluation of Architecture-centric RSRs (§ 5.1). ○, △, and × indicate a property is fully satisfied, partially satisfied, and unsatisfied, respectively.

Category		Method	P1 Multi-cause Robustness	P2 Dual-manifestation Robustness	P3 Dual-phase Robustness	P4 Motivation Transformation Awareness	P5 Generality	P6 Data Accessibility	P7 Scalability	P8 Theoretical Grounding
Attention Mechanism (§ 5.1.1)	Basic Attention	NARM [58]	△	△	△	×	×	○	△	×
		STAMP [59]	△	△	△	×	×	○	○	×
	Self Attention	SASRec [23]	△	△	△	×	×	○	△	×
		BERT4Rec [61]	△	△	△	×	×	○	△	×
		DT4Rec [62]	△	△	△	×	×	○	△	×
		STOSA [43]	△	△	△	×	×	○	△	×
		ADT [63]	△	△	△	×	×	○	△	×
		AC-TSR [64]	△	△	△	×	×	○	△	×
	Sparse Attention	DSAN [65]	△	△	△	×	×	○	△	×
		Locker [67]	△	△	△	×	×	○	△	×
		RecDenoiser [68]	△	△	△	×	×	○	△	×
		RETR [69]	△	△	△	×	×	○	△	×
		DPDM [70]	△	△	△	×	×	○	△	×
		AutoDisenSeq [71]	△	△	△	×	×	○	△	×
	Knowledge-enhanced Attention	IARN [72]	△	△	△	×	×	×	×	×
		NOVA [73]	△	△	△	×	×	×	×	×
		KGDPL [74]	△	△	△	×	×	×	×	×
Memory Networks (§ 5.1.2)	Key-value MN	RUM [76]	△	△	△	×	×	○	×	×
		MAGNN [77]	△	△	△	×	×	○	×	×
		DMAN [78]	△	△	△	×	×	○	×	×
		MASR [79]	△	△	△	×	×	○	△	×
	Knowledge-enhanced MN	KSR [80]	△	△	△	×	×	×	×	×
		CmnRec [81]	△	△	△	×	×	×	△	×
		LMN [82]	△	△	△	×	×	×	△	×
Gating Networks (§ 5.1.3)	Basic Gating Networks	HGN [84]	△	△	△	×	×	○	△	×
		M3R [85]	△	△	△	×	×	○	△	×
		ASPPA [86]	△	△	△	×	×	○	△	×
		STAR-Rec [87]	△	△	△	×	×	○	△	×
	Consistency-aware Gating Networks	π-Net [88]	△	△	△	×	×	○	△	×
		CAR [89]	△	△	△	×	×	○	△	×
		MAN [91]	△	△	△	×	×	○	△	×
		S2PNM [90]	△	△	△	×	×	○	△	×
Graph Neural Networks (§ 5.1.4)	Basic GNNs	GCSAN [?]	△	△	△	×	×	○	△	×
		SRGNN [?]	△	△	△	×	×	○	△	×
	Knowledge-enhanced GNNs	IMFOU [93]	△	△	△	×	×	×	×	×
		FAPAT [94]	△	△	△	×	×	×	×	×
		I-DIDA [96]	△	△	△	△	×	×	×	×
		TGODE [95]	△	△	△	△	×	×	×	×
	Sparse GNNs	SLED [97]	△	△	△	×	×	○	△	×
		MAERec [98]	△	△	△	×	×	○	△	×
		GDRN [99]	△	△	△	×	×	○	△	×
		RAIN [100]	△	△	△	×	×	○	△	×
		MA-GCL4SR [101]	△	△	△	×	×	○	△	×
Time-frequency Analysis (§ 5.1.5)	Fourier Transform	FMLP [103]	△	△	△	×	×	○	○	×
		SLIME4Rec [104]	△	△	△	×	×	○	△	×
		BSARec [105]	△	△	△	×	×	○	△	×
		FDCLRec [106]	△	△	△	×	×	○	△	×
		LFDFSR [107]	△	△	△	×	×	×	△	×
		END4Rec [108]	△	△	△	×	×	○	△	×
		DPCPL [109]	△	△	△	×	×	○	△	×
		SSR [110]	△	△	△	×	×	○	○	×
		Oracle4Rec [111]	△	△	△	×	×	○	△	×
		DIFF [112]	△	△	△	×	×	×	△	×
	Wavelet Transform	Wavelet [113]	△	△	△	×	×	○	○	×
		WaveRec [114]	△	△	△	×	×	○	○	×
		WEARec [115]	△	△	△	×	×	○	○	×
	Hybrid Frequency Attention	FEARec [116]	△	△	△	×	×	○	△	×
		MUFFIN [117]	△	△	△	×	×	○	△	×
		FICLRec [118]	△	△	△	×	×	○	△	×
		MIRRN [119]	△	△	△	×	×	○	×	×
Diffusion Models (§ 5.1.6)	Basic Diffusion	DiffuRec [120]	△	△	△	×	×	○	×	×
		CF-Diff [121]	△	△	△	×	×	○	×	×
	Conditional Diffusion	CDDRec [122]	△	△	△	×	×	○	×	×
		SeeDRec [123]	△	△	△	×	×	×	×	×
		M3BSR [124]	△	△	△	×	×	×	×	×
		TDM [125]	△	△	△	×	×	○	×	×
	Advanced Diffusion	PDRec [126]	△	△	△	×	×	○	×	×
		FMRec [127]	△	△	△	×	×	○	×	×
		ADRec [128]	△	△	△	×	×	○	×	×
		DiffDiv [129]	△	△	△	×	×	○	×	×
		DiQDiff [130]	△	△	△	×	×	○	×	×

5.1.6. Architecture-Centric RSRs Based on Diffusion Models

Diffusion models enhance robustness via a progressive denoising process, learning to reconstruct preference-consistent representations from corrupted inputs and extract resilient preference patterns from perturbed sequences. Existing methods can be categorized into *Basic Diffusion*, *Conditional Diffusion*, and *Advanced Diffusion*.

 doi:10.20944/preprints202601.2218.v1

Subgroup 1: Basic Diffusion uses denoising diffusion to recover robust item/sequence representations. The forward process gradually corrupts the target item embedding $\mathbf{v}_t^{u,0}$:

$$q(\mathbf{v}_t^{u,k}|\mathbf{v}_t^{u,k-1}) = \mathcal{N}(\mathbf{v}_t^{u,k}; \sqrt{1-\beta_k}\mathbf{v}_t^{u,k-1}, \beta_k\mathbf{I}), \tag{15}$$

where k is the diffusion step; $\mathcal{N}(x; \mu, \sigma^2)$ is a Gaussian distribution with a mean $\sqrt{1-\beta_k}$ and variance β_k, which is generated from a pre-defined noise schedule. The reverse process reconstructs the target item embedding conditioned on the input sequence $\mathbf{x}_t^u$:

$$p_\theta(\mathbf{v}_t^{u,k-1}|\mathbf{v}_t^{u,k}, \mathbf{c}) = \mathcal{N}(\mathbf{v}_t^{u,k-1}; \mu_\theta(\mathbf{v}_t^{u,k}, \mathbf{c}, k), \sigma_k^2\mathbf{I}), \tag{16}$$

where $\mathbf{c}$ is the sequence-encoded context embedding, $\mu_\theta(\cdot)$ and σ_t^2 are the mean and variance of the denoised distribution parameterized by θ. The model is trained to minimize the reconstruction loss between the predicted and true target embeddings. Specifically, **DiffuRec** [120] models items as Gaussian distributions and uses a diffusion-reverse framework to generate distributional representations for historical items, enhancing robustness to perturbations via stochastic noise injection. **CF-Diff** [121] integrates high-order user-item graph connectivity into the diffusion process via a cross-attention-guided multi-hop autoencoder, leveraging neighbor signals to filter perturbations.

Subgroup 2: Conditional Diffusion guides denoising with auxiliary conditions (attributes, semantics, multi-modal signals) by integrating conditions into $\mathbf{c}$ during the reverse process. For example, **CDDRec** [122] uses a conditional diffusion framework with a cross-attentive decoder to generate high-fidelity user preferences, reducing perturbation-induced oversmoothing. **SeeDRec** [123] operating on sememes (minimal interest units) rather than item indices, using Sememe-to-Interest diffusion model to capture generalized interest distributions that are less sensitive to perturbations. **M3BSR** [124] incorporates multi-modal/multi-behavior data, denoising modalities/behaviors separately via conditional diffusion and using ID features/favor behaviors to guide denoising. **TDM** [125] simulates missing items via dual-side Thompson sampling, conditioning diffusion on edited sequences to learn consistency under perturbations.

Subgroup 3: Advanced Diffusion introduces advanced adaptations (flow matching, auto-regressive training, quantization) to address data sparsity, embedding collapse, or diversity challenges. For example, **PDRec** [126] uses a pre-trained diffusion model to estimate user preferences for all items, reweighting historical behaviors and augmenting samples to address data sparsity and perturbations. **FMRec** [127] replaces stochastic diffusion with flow matching, using deterministic reverse sampling to reduce error accumulation and eliminate recommendation randomness. **ADRec** [128] addresses embedding collapse via token-level diffusion (independent diffusion for each sequence token), enabling auto-regressive learning and per-token distribution modeling. **DiffDiv** [129] introduces diversity-aware guidance learning, conditioning diffusion on sampled guidance signals to generate diverse and accurate recommendations. **DiQDiff** [130] uses semantic vector quantization to discretize sequences into semantic codes, combining contrastive discrepancy maximization to ensure personalized generation against perturbations.

Discussion and Assessment (detailed in Table 2). Diffusion-based RSRs effectively handle behavioral randomness via iterative perturbation injection and denoising, addressing partial mismatch and enhancing training-phase robustness. Yet, they lack mechanisms for contextual influences, malicious manipulations, complete mismatch, motivational transformations, and inference-phase robustness. They require architectural adaptation, with conditional and advanced variants demanding auxiliary data. Scalability is limited by iterative denoising—especially for long sequences and large item catalogs—and formal theoretical guarantees on robustness are absent.

5.2. Data-Centric RSRs

Data-centric RSRs enhance robustness by intervening at the instance construction stage—where mismatched input-target pairs first emerge—with the core goal of shielding model learning from unreliable instances. They achieve this by proactively filtering, correcting, or supplementing problematic pairs, mitigating perturbations at the source to reduce the model's exposure to erroneous sequential patterns. Existing methods are categorized into three primary groups: *Instance Selection*, *Instance*

 doi:10.20944/preprints202601.2218.v1

Correction, and *Instance Augmentation*. Figure 7 outlines this paradigm's evolution, with detailed methodological breakdowns below.

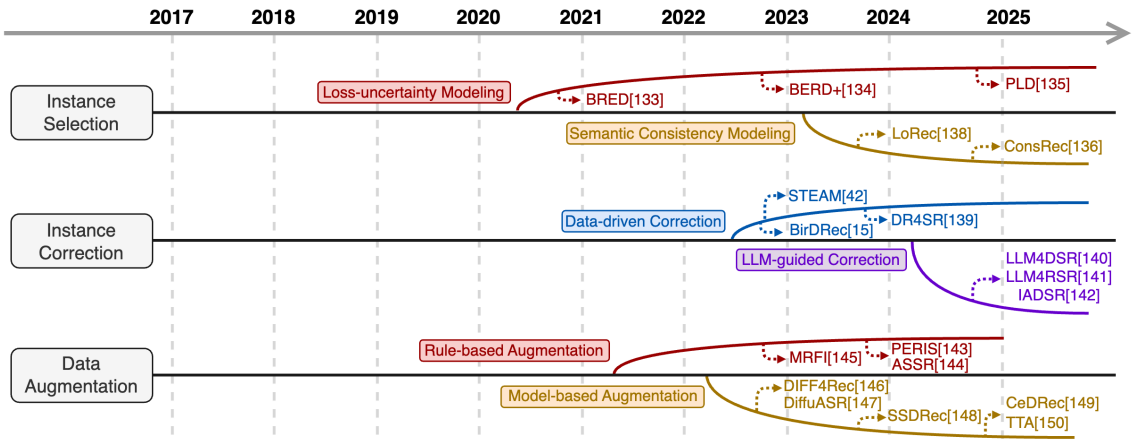

Figure 7. Development trajectory of Data-centric RSRs.

5.2.1. Data-Centric RSRs Based on Instance Selection

Instance selection methods clean training data by identifying and discarding/reweighting mismatched input-target pairs, preventing erroneous patterns from entering training. Their objective function is formalized as:

$$\mathcal{L}_{\text{select}} = \sum_{\langle x_u^t, v_u^t \rangle \in \mathcal{D}_{\text{train}}} \text{rel}(x_u^t, v_u^t) \cdot \phi(f(x_t^u, v_t^u)) \tag{17}$$

where $\text{rel}(x_u^t, v_u^t) \in [0,1]$ denotes the estimated reliability score of each instance, and $\phi(\cdot)$ is the base recommendation loss function (Equation 1). Existing Instance-selection-based RSRs are classified into two subgroups: *Loss-uncertainty Modeling* and *Semantic Consistency Modeling*.

Subgroup 1: Loss-uncertainty Modeling estimates instance reliability by jointly modeling instance loss and uncertainty:

$$\text{rel}(x_t^u, v_t^u) = 1 - \mathbb{I}(\phi_{u,t} > \tau_1 \wedge h_{u,t} < \tau_2), \tag{18}$$

where $\phi_{u,t}$ and $h_{u,t}$ are the loss and uncertainty of the instance, $\mathbb{I}(\cdot)$ is the indicator function, and τ_1/τ_2 are thresholds. This formulation assumes that unreliable instances typically exhibit high loss but low uncertainty, while valuable uncertain instances (e.g., ambiguous patterns) have high loss and high uncertainty. In particular, **BERD** [131] Pioneers the approach by modeling uncertainty as Gaussian entropy, filtering high-loss/low-uncertainty instances. **BERD+** [132] Extends BERD with item attributes via heterogeneous graph convolution, correcting estimates for cold items. **PLD** [133] uses fine-grained personalized loss distributions to identify unreliable instances.

Subgroup 2: Semantic Modeling leverages semantic consistency between input and targets to calculate instance reliability, assuming interactions driven by intrinsic preferences form semantically coherent clusters while perturbations are outliers. In particular, **ConsRec** [134] constructs item graphs based on T5 [135]-encoded semantic similarity, filtering perturbations via user-consistent preference modeling: $s_{u,i} = \frac{1}{|C_u|} \sum_{v_j \in C_{\max}} \text{sim}(\mathbf{v}_i, \mathbf{v}_j)$, where $s_{u,i}$ is the semantic consistency between item v_i and user u's preference; C_u is the maximum connected component in the user u's interaction graph; $\mathbf{v}_i$ is the item embedding encoded by the language model T5. **LoRec** [136] uses an LLM-enhanced calibrator to estimate fraudster likelihood, downweighting malicious instances via iterative weight compensation.

Discussion and Assessment (detailed in Table 3). Selection-based RSRs enhance training-phase robustness: loss-uncertainty methods address behavioral randomness, LoRec excels against malicious manipulations, and most target complete mismatch (with ConsRec handling partial mismatch). However, they neglect inference-phase robustness and motivational transformations. Practically, they are plug-and-play for existing SRSs, with semantic consistency variants demanding auxiliary features. Scalability is constrained by latency from language model integration in semantic methods, and formal robustness proofs are absent.

 doi:10.20944/preprints202601.2218.v1

5.2.2. Data-Centric RSRs Based on Instance Correction

Instance correction methods modify unreliable instances to form matched input-target pairs, avoiding data sparsity caused by outright removal. Their objective function is:

$$\mathcal{L}_{\text{correct}} = \sum_{\langle \mathbf{x}_u^t, v_u^t \rangle \in \mathcal{D}_{\text{train}}} \phi(f(\text{correct}(\mathbf{x}_t^u, v_t^u))), \tag{19}$$

where $\text{correct}(\cdot,\cdot)$ is a corrector that rectifies mismatched pairs. Existing instance-correction-based RSRs are classified into *Data-driven Correction* and *LLM-guided Correction*.

Subgroup 1: Data-driven Correction leverages intrinsic interaction patterns (e.g., co-occurrence frequencies, reconstruction errors) to rectify unreliable instances without external knowledge. Specifically, **STEAM** [42] designs a self-correcting framework with item-wise 'keep/delete/insert' operations, trained via self-supervised perturbation simulation. **BirDRec** [15] proposes bidirectional rectification (forward SRS replaces low-score targets; backward SRS removes low-score inputs) with theoretical error bounds. **DR4SR** [137] regenerates reliable and informative training data via a diversity-promoted regenerator, balancing exploitation/exploration for generalizability.

Subgroup 2: LLM-guided Correction harnesses the semantic reasoning and open-world knowledge of LLMs to enhance perturbation identification and correction, addressing limitations of data-driven methods in handling inactive users and cold items. The LLM-guided corrector can be defined as:

$$\text{correct}(\mathbf{x}_t^u, v_t^u) = \text{LLM}(\text{prompt}(\mathbf{x}_t^u, v_t^u)), \tag{20}$$

where $\text{prompt}(\cdot)$ constructs task-specific prompts for the LLM. Specifically, **LLM4DSR** [138] Fine-tunes LLMs with self-supervised instruction tuning to identify inconsistent input items, filtering low-confidence corrections. **LLM4RSR** [139] uses textual gradient descent to optimize prompts for instance correction. **IADSR** [140] aligns LLM semantic embeddings with collaborative embeddings, masking perturbations via multi-perspective consistency scores.

Discussion and Assessment (detailed in Table 3) Correction-based RSRs excel against behavioral randomness, effectively addressing both partial and complete mismatch while enhancing training-phase robustness. They are highly model-agnostic: data-driven methods rely solely on interaction data and are computationally efficient, while LLM-guided variants demand auxiliary item text and incur high overhead. However, they lack mechanisms to counter malicious attacks, neglect inference-phase robustness and motivational transformations, and most methods lack formal theoretical guarantees.

Table 3. Evaluation of Data-centric RSRs (§ 5.2). ○, △, and × indicate a property is fully satisfied, partially satisfied, and unsatisfied, respectively.

Category		Method	P1 Multi-cause Robustness	P2 Dual-manifestation Robustness	P3 Dual-phase Robustness	P4 Motivation Transformation Awareness	P5 Generality	P6 Data Accessibility	P7 Scalability	P8 Theoretical Grounding
Instance Selection (§ 5.2.1)	Loss-uncertainty Modeling	BERD [131]	△	△	△	×	○	○	△	×
		BERD+ [132]	△	△	△	×	○	×	△	×
		PLD [133]	△	△	△	×	○	○	○	×
	Semantic Modeling	LoRec [136]	△	△	△	×	○	×	×	×
		ConsRec [134]	△	△	△	×	×	×	×	×
Instance Correction (§ 5.2.2)	Data-driven Correction	STEAM [42]	△	○	△	×	○	○	×	×
		BirDRec [15]	△	○	△	×	○	○	△	○
		DR4SR [137]	△	○	△	×	○	○	△	×
	LLM-guided Correction	LLM4DSR [138]	△	△	△	×	○	×	×	×
		LLM4RSR [139]	△	○	△	×	○	×	△	×
		IADSR [140]	△	△	△	×	○	△	×	×
Data Augmentation (§ 5.2.3)	Rule-based Augmentation	PERIS [141]	△	△	△	×	○	△	○	×
		MRFI [143]	△	△	△	×	○	○	○	×
		ASSR [142]	△	△	△	×	○	○	△	×
	Model-base Augmentation	Diff4Rec [144]	△	△	△	×	○	○	×	×
		DiffuASR [145]	△	△	△	×	○	○	×	×
		SSDRec [146]	△	△	△	×	○	○	×	×
		CeDRec [147]	△	△	△	×	×	○	×	×
		TTA [148]	△	△	△	×	○	○	○	×

5.2.3. Data-Centric RSRs based on Data Augmentation

Data augmentation dilutes unreliable instances by generating semantically consistent synthetic data, with the objective function:

$$\mathcal{L}_{\text{aug}} = \sum_{\langle \mathbf{x}_u^t, v_u^t \rangle \in \mathcal{D}_{\text{train}}} [\phi(f(\mathbf{x}_u^t, v_u^t)) + \lambda \cdot \phi(f(\text{aug}(\mathbf{x}_u^t, v_u^t)))] \tag{21}$$

 doi:10.20944/preprints202601.2218.v1

where $\text{aug}(\cdot,\cdot)$ is the augmentation operation applied to an instance, and λ controls the contribution of augmented data. Based on the techniques for implementing the augmentation function, existing methods can be classified into *Rule-based Augmentation* and *Model-based Augmentation*.

Subgroup 1: Rule-based Augmentation leverages hand-crafted or mined association rules to generate reliable instances. Specifically, **PERIS** [141] augments via similar items and like-minded users' behaviors. It mines users' temporal consumption patterns to model interest sustainability and filters perturbations in the original and augmented data. **ASSR** [142] inserts contextually relevant items from frequent item sets, with a confidence mechanism to filter low-quality augmentations. **MRFI** [143] reformulates the objective to consider multiple future targets, reducing over-reliance on perturbed targets via relevance-aware loss.

Subgroup 2: Model-based Augmentation employs deep generative models to create high-quality augmented data that capture underlying user preference distributions. Specifically, **Diff4Rec** [144] and **DiffuASR** [145] adopt diffusion models for sequence generation, with forward corruption and reverse denoising to produce reliable augmentations. **SSDRec** [146] proposes a self-augmentor that unifies item-level/subsequence-level augmentation, with hierarchical denoising to avoid over/under-denoising. **CeDRec** [147] generates contrastive views via hierarchical augmentation, using an adaptive expert network to focus on critical patterns. **TTA** [148] enhances inference-phase robustness via test-time augmentation (TMask/TNoise) and prediction aggregation, introducing controlled perturbations while preserving sequential patterns.

Discussion and Assessment (detailed in Table 3). Augmentation-based RSRs handle behavioral randomness effectively, addressing both partial and complete mismatch by diluting unreliable instances while enhancing training-phase robustness (with TTA boosting inference-phase robustness). However, they lack defenses against malicious manipulations, overlook motivational transformations, and are deeply coupled with model architectures. Practically, rule-based methods rely solely on interaction data, while some model-based variants demand auxiliary signals. Scalability varies: simple rule-based operations are efficient, but complex generative models incur high computational costs, and formal robustness guarantees are absent.

5.3. Learning-Centric RSRs

Learning-centric RSRs enhance robustness by optimizing the model training process, without modifying the core architecture or raw data. They adopt robustness-oriented training strategies—including *Adversarial Training*, *Reinforcement Learning*, *Distributionally Robust Optimization*, *Self-supervised Learning*, *Causal Learning*, and *Curriculum Learning*—to suppress the interference of unreliable instances during training. Figure 8 illustrates the development of this paradigm, with detailed methodological groups below.

 doi:10.20944/preprints202601.2218.v1

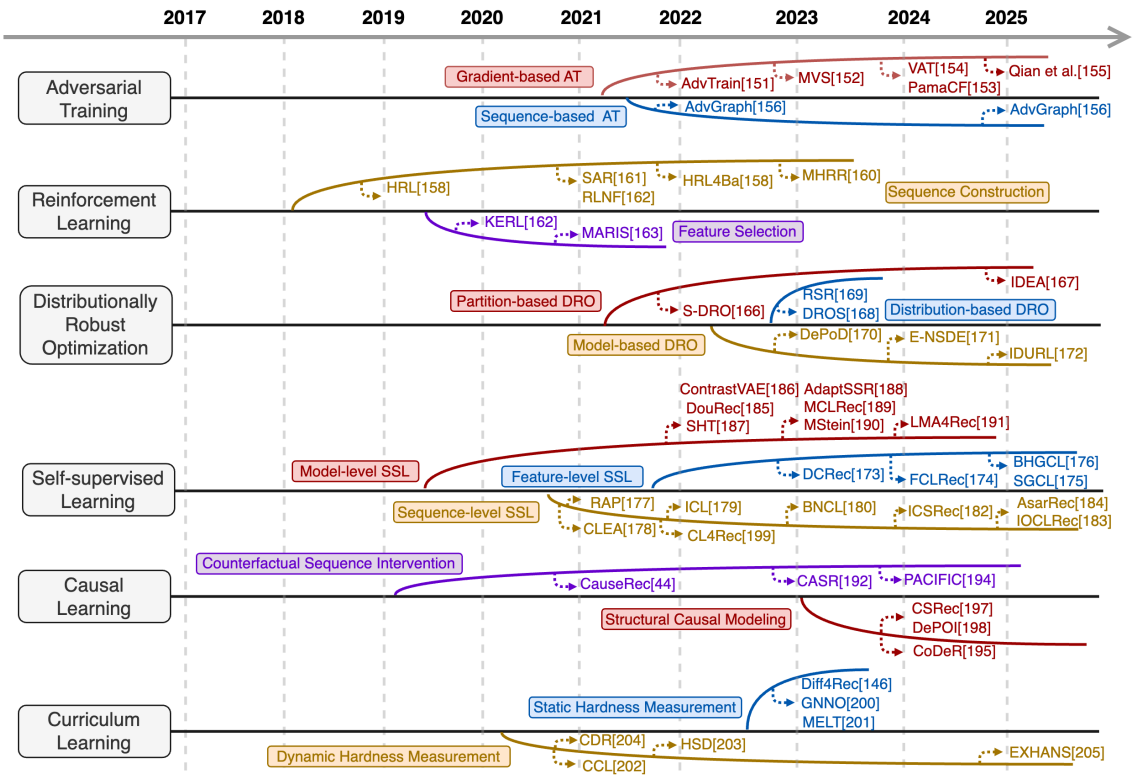

Figure 8. Development trajectory of Learning-centric RSRs.

5.3.1. Learning-Centric RSRs Based on Adversarial Training

Adversarial Training (AT) strengthens SRS robustness by explicitly exposing the model to elaborately designed perturbations during training. It formulates a minimax game between a perturbation generator and a recommender. By optimizing the model's worst-case performance, AT forces the model to learn perturbation-invariant preference patterns, improving resilience to unreliable instances. Existing methods are categorized into *Gradient-Based AT* and *Sequence-based AT*.

Subgroup 1: Gradient-Based AT generates perturbations using gradient information to attack model parameters or embeddings, targeting vulnerabilities in the model's parameter space. The objective is formalized as:

$$\min_{\Theta} \mathbb{E}_{(x_t^u, v_t^u)\sim\mathcal{D}_{\text{train}}} \left[\max_{\Delta\in\mathcal{B}(\epsilon)} \phi(f(x_t^u, v_t^u; \Theta+\Delta)) \right] \tag{22}$$

where Θ is the parameters of recommender f, Δ represents the adversarial perturbation, and $\mathcal{B}(\epsilon)$ is a constraint set (e.g. $||\Delta|| \leq \epsilon$) ensuring perturbations are bounded. This forces the model to learn parameters that are stable under perturbations. Specifically, **AdvTrain** [149] uses Dirichlet neighborhood sampling and item embedding perturbation to defend against malicious substitutions. **MVS** [150] enhances item embeddings with combinations of similar items predicted by a complementary model, incorporating gradient-based noise to mimic behavioral randomness. **PamaCF** [151] personalizes perturbation magnitudes based on user embedding norms, tailoring defenses against poisoning attacks. **VAT** [152] scales perturbations by user training loss, focusing defenses on vulnerable users. **Qian et al.** [153] injects noise into model weights, emulating denoising autoencoders for prediction stability.

Subgroup 2: Sequence-based AT perturbs user interaction sequences to encourage invariance, enhancing robustness through sequence-level manipulations. The objective function is:

$$\mathcal{L}_{\text{seq\_adv}} = \sum_{\langle x_u^t, v_u^t\rangle\in\mathcal{D}_{\text{train}}} \left[\phi(f(x_u^t, v_u^t)) - \lambda \cdot \sum_{m=1}^{M} \phi(f(\text{perturb}(x_u^t, v_u^t))) \right], \tag{23}$$

where $\text{perturb}(\cdot,\cdot)$ manipulates the original instance via item additions, replacements, or removals. Specifically, **AdvGraph** [154] applies adversarial perturbations to the graph structure of user-item interactions, using a reinforce-based gradient estimator to optimize discrete edge changes for improved

 doi:10.20944/preprints202601.2218.v1

robustness. **DARTS** [155] combines explicit item substitution and implicit contrastive learning, crafting poisoned gradients to boost target item exposure under limited malicious clients.

Discussion and Assessment (detailed in Table 4). Adversarial Training-based RSRs excel in handling malicious manipulations via perturbation injection, effectively addressing partial mismatch and enhancing training-phase robustness. However, they neglect behavioral randomness, contextual influences, complete mismatch, inference-phase robustness, and motivational transformations. Practically, they are model-agnostic with only minor loss function adaptations, and most rely solely on user-item interactions. Scalability is constrained by iterative training, and formal robustness guarantees are absent.

5.3.2. Learning-Centric RSRs Based on Reinforcement Learning

Reinforcement Learning (RL) formulates sequential recommendation as a sequential decision-making problem, where an agent optimizes long-term user engagement. In RSRs, RL enables the model to treat perturbations as suboptimal signals and to suppress them by learning policies that reward stable preference patterns. Existing RL-based RSRs can be classified into two subgroups: *Adaptive Sequence Construction* and *Auxiliary Information Selection*.

Subgroup 1: Adaptive Sequence Construction uses RL to dynamically adjust input sequence construction (e.g., truncation, item filtering) to mitigate perturbations. The process is modeled as a Markov Decision Process (MDP), where a policy $\pi(a_t|s_t)$ learns to modify the input sequence $\mathbf{x}_t^u$ to enhance recommendation accuracy. The state s_t represents the user's current sequence, the action a_t involves decisions like truncating the sequence or removing items, and the reward $R(a_t, s_t)$ is based on the improvement in predicting the target item v_t^u, such as the log-likelihood gain after adaptation. The policy $\pi(a_t|s_t)$ is learned to maximize:

$$\sum_{\langle \mathbf{x}_t^u, v_t^u \rangle \in \mathcal{D}_{\text{train}}} \log \pi(a_t|s_t = \mathbf{x}_t^u) R(a_t, s_t = \mathbf{x}_t^u). \tag{24}$$

Specifically, **HRL** [156], **HRL4Ba** [157], and **MHRR** [158] employ hierarchical RL agents to revise user profiles—high-level policies decide whether to revise sequences, and low-level policies filter irrelevant items. **SAR** [159] uses actor-critic RL to adaptively select sequence lengths for each user, truncating noisy suffixes to exclude perturbations. **RLNF** [160] models false negative sample filtering as an RL problem, selecting reliable negatives to reduce label perturbations.

Subgroup 2: Auxiliary Information Selection leverages RL to select relevant auxiliary information (e.g., knowledge graphs, textual features), avoiding noisy feature integration that distorts sequential patterns. The state s_t includes the sequence and auxiliary features, while the action a_t selects feature subsets to maximize the reward $R(a_t, s_t)$, which is measured by the improvement in recommendation quality after feature integration. Specifically, **KERL** [161] Integrates knowledge graphs into RL, with states incorporating current and predicted future preferences, and rewards combining sequence-level and knowledge-level metrics to capture long-term preference drifts. **MARIS** [162] uses multi-agent RL for feature selection, with a QMIX network coordinating agents to produce optimal feature combinations.

Discussion and Assessment (detailed in Table 4). RL-based RSRs handle behavioral randomness effectively via long-term reward optimization, excelling at partial mismatch through adaptive filtering and enhancing training-phase robustness. However, they overlook contextual influences, malicious attacks, complete mismatch, motivational transformations, and inference-phase robustness. They require significant architectural modifications. Methods for feature selection demand auxiliary information. Scalability is constrained by high computational costs of RL training, and formal robustness guarantees are absent.

5.3.3. Learning-Centric RSRs Based on Distributionally Robust Optimization

Distributionally Robust Optimization (DRO) [163] offers a theoretically rigorous framework for robustness, optimizing model performance under the worst-case distribution within a predefined uncertainty set. For sequential recommendation, DRO addresses distribution shifts caused by behavioral randomness and contextual influences, ensuring recommendations remain reliable when the

 doi:10.20944/preprints202601.2218.v1

training and test distributions diverge. The uncertainty set is defined via divergence metrics (e.g., KL-divergence, Wasserstein distance), capturing perturbations like preference drift. Existing methods can be classified into *Partition-based DRO*, *Distribution-based DRO*, and *Model-based DRO*.

Subgroup 1: Partition-based DRO partitions users/sequences into groups and optimizes for the worst-case performance across groups, addressing distribution shifts from behavioral randomness or contextual influences. The objective is to minimize the maximum loss over groups:

$$\arg\min_{\Theta}\left\{\max_{w_g\in\Delta_m}\sum_{g=1}^{G} w_g\mathbb{E}_{(x_t^u,v_t^u)\sim\mathcal{D}_g}[\phi(f(\mathbf{x}_t^u,v_t^u;\Theta))]\right\}, \tag{25}$$

where $\mathcal{D}_g$ denotes the data distribution for group g, w_g is the weight for group g, Δ_G is the $(G-1)$-dimensional probability simplex. This formulation ensures that the model performs well even for disadvantaged subgroups affected by perturbations. Specifically, **S-DRO** [164] dynamically reweights training instances to prioritize groups with high losses (associated with unreliable instances), ensuring robust pattern learning across diverse user behaviors. **IDEA** [165] generates diverse training environments via item dropping and mixup augmentation, using DRO and invariant risk minimization to learn stable preferences.

Subgroup 2: Distribution-based DRO defines an uncertainty set around a nominal distribution (estimated from historical actions), minimizing the worst-case loss over distributions within the set:

$$\arg\min_{\Theta}\left\{\max_{Q\in\mathcal{Q}}\mathbb{E}_{(x_t^u,v_t^u)\sim Q}[\phi(f(\mathbf{x}_t^u,v_t^u;\Theta))]\right\}, \tag{26}$$

where $\mathcal{Q}=\{Q:\mathrm{div}(Q,\mu_0)\le\rho\}$ is the uncertainty set, u_0 is the nominal distribution estimated from historical data, $\mathrm{div}(\cdot,\cdot)$ is a divergence measure (e.g., KL-divergence), and ρ is the robust radius. In particular **DROS** [166] uses KL-divergence to define the uncertainty set, accounting for gradual distribution shifts (e.g., motivation transformation) to generalize to unseen test distributions. **RSR** [167] groups sequences into clusters, minimizing the worst-case loss over clusters to enhance robustness to distribution shifts.

Subgroup 3: Model-based DRO integrates DRO principles into the model architecture via multi-model distillation or uncertainty quantification. Specifically, **DePoD** [168] decouples knowledge distillation for items across multiple peer networks. By minimizing the divergence between the student's predictions and the teachers' consensus, the model prioritizes features that are consistent across different models, which are more likely to represent genuine user preferences rather than perturbations. **E-NSDE** [169] combines neural stochastic differential equations with evidential learning to quantify uncertainty that grows with time intervals, guiding exploration for long-term preferences and reducing errors from interest shifts. **IDURL** [170] quantifies interest drift magnitude using category information and disentangles user representations into multiple drift levels, aligned via collaborative signals.

Discussion and Assessment (detailed in Table 4). DRO-based RSRs address behavioral randomness and contextual influences via distributional shift modeling, excelling at addressing partial mismatch and motivation transformations, and enhancing training-phase robustness. However, they lack mechanisms against malicious attacks, struggle with complete mismatch, and overlook inference-phase robustness. Practically, they are model-agnostic and rely solely on interaction data, while model-based variants incur higher computational costs. Notably, they implicitly account for motivational temporal evolution, and the paradigm inherently offers theoretical foundations.

5.3.4. Learning-Centric RSRs Based on Self-supervised Learning

Self-supervised Learning (SSL) enhances SRS robustness by constructing auxiliary supervisory signals from raw interaction data. Unlike supervised learning (relying on next-item prediction), SSL constructs positive/negative views (contrastive pairs) to distinguish genuine preferences from perturbations. The core of SSL-based RSRs lies in view construction—aligning semantically consistent views while pushing apart perturbed ones. Existing methods can be categorized into three subgroups: *Feature-level SSL*, *Sequence-level SSL*, and *Model-level SSL*.

 doi:10.20944/preprints202601.2218.v1

Subgroup 1: Feature-level SSL constructs positive views via semantic/structural feature similarity, ensuring positive signals are grounded in meaningful attribute correlations rather than extrinsically motivated co-occurrences. The contrastive loss is:

$$\mathcal{L}_{\text{feat-SSL}} = -\sum_{<\mathbf{x}_t^u, \mathbf{v}_t^u> \in \mathcal{D}_{\text{train}}} \log \frac{\exp(\text{sim}(\text{encode}_{\text{feat}}(\mathbf{x}_t^u), \text{encode}_{\text{feat}}^{+}(\mathbf{x}_t^u))/\tau_3)}{\sum_{\mathbf{x}' \in \mathcal{N}_{u,t}^{-}} \exp(\text{sim}(\text{encode}_{\text{feat}}(\mathbf{x}_t^u), \text{encode}_{\text{feat}}(\mathbf{x}'))/\tau_3)}, \tag{27}$$

where $\text{encode}_{\text{feat}}(\cdot)$ is the feature-aware encoder that fuses item features with sequential patterns, $\text{encode}_{\text{feat}}^{+}(\cdot)$ generates feature-manipulated positive views. $\mathcal{N}_{u,t}^{-}$ is the set of negative views (sequences with dissimilar features), and τ_3 is the temperature parameter. Specifically, **DCRec** [171] integrates item transition/co-interaction graphs to enrich representations and estimate interaction conformity, adaptively down-weighting contrastive signals from low-conformity (perturbation-prone) interactions. **FCLRec** [172] proposes bidirectional feature-aware self-attention, constructing positive views via feature masking to ensure invariance to minor feature perturbations. **SGCL** [173] uses symmetric contrastive loss to align risk minimizers under perturbed and clean data, enhancing tolerance to feature-level perturbations by suppressing semantically inconsistent views. **BHGCL** [174] combines the information bottleneck principle with feature-level contrastive learning, pruning perturbation-prone edges while preserving minimal sufficient features.

Subgroup 2: Sequence-level SSL generates positive views via moderate sequence manipulations (masking, cropping, reordering), retaining core sequential patterns while introducing controlled perturbations. The contrastive loss is:

$$\mathcal{L}_{\text{seq-SSL}} = -\sum_{<\mathbf{x}_t^u, \mathbf{v}_t^u> \in \mathcal{D}_{\text{train}}} \log \frac{\exp(\text{sim}(\text{encode}_{\text{seq}}(\mathbf{x}_t^u), \text{encode}_{\text{seq}}^{+}(\mathbf{x}_t^u))/\tau_3)}{\sum_{\mathbf{x}' \in \mathcal{N}_{u,t}^{-}} \exp(\text{sim}(\text{encode}_{\text{seq}}(\mathbf{x}_t^u), \text{encode}_{\text{seq}}(\mathbf{x}'))/\tau_3)}, \tag{28}$$

where $\text{encode}_{\text{seq}}$ and $\text{encode}_{\text{seq}}^{+}$ denote sequence encoders (e.g., Transformer), with $\text{encode}_{\text{seq}}^{+}$ additionally manipulating the original sequence to generate positive views. Specifically, **RAP** [175] uses a policy network to select relevant items and discard perturbations, maximizing similarity between positive subsequences and targets. **CLEA** [176] splits baskets into positive/negative sub-baskets via Gumbel-Softmax, aligning positive sub-baskets with targets via anchor-guided contrastive loss. **ICL** [177] clusters sequence representations into intent prototypes (positive views) via Expectation-Maximization (EM), reducing sensitivity to perturbations with invariant cross-user intent patterns. **BNCL** [178] constructs positive views by applying consistency-aware augmentation (dropping low-importance edges) to user-item graphs. **CL4SRec** [179] constructs positive views via sequence crop, mask, and reorder, training the encoder to ignore minor perturbations and focus on core sequential patterns. **ICSRec** [180] and **IOCLRec** [181] segment sequences into intent-consistent subsequences as positive views, filtering perturbation-prone items misaligned with target intent. **AsarRec** [182] proposes an adaptive sequence manipulation strategy tailored for individual users and specific contexts, thereby accommodating diverse scenarios.

Subgroup 3: Model-level SSL generates positive views by moderately manipulating model parameters (e.g., dropout masks) rather than input sequences, training the model to produce invariant representations resilient to perturbations. The contrastive loss is:

$$\mathcal{L}_{\text{model-SSL}} = -\sum_{<\mathbf{x}_t^u, \mathbf{v}_t^u> \in \mathcal{D}_{\text{train}}} \log \frac{\exp(\text{sim}(\mathbf{h}_{\theta_1}(\mathbf{x}_t^u), \mathbf{h}_{\theta_2}(\mathbf{x}_t^u))/\tau)}{\sum_{\mathbf{x}' \in \mathcal{N}_{u,t}^{-}} \exp(\text{sim}(\mathbf{h}_{\theta_1}(\mathbf{x}_t^u), \mathbf{h}_{\theta_1}(\mathbf{x}'))/\tau)}, \tag{29}$$

where θ_1 and θ_2 are perturbed parameters, $\mathbf{h}_\theta(\cdot)$ denotes the model's representation output with parameters θ. In particular, **DuoRec** [183] and **ContrastVAE** [184] uses different dropout masks to generate positive views, reshaping embedding distributions to reduce perturbation sensitivity. **SHT** [185] captures global parameter-invariant patterns via hypergraph propagation, generating edge 'solidity scores' to identify perturbations. **AdaptSSR** [186] generates positive views via implicit (dropout) and explicit (mask/crop) model manipulations, then replaces strict contrastive consistency with self-supervised ranking (implicit views > explicit views > other users' views), dynamically adjusting loss weights for low-quality views. **MCLRec** [187] uses meta-optimized learnable augmenters to generate high-quality positive views, combining data- and model-level contrastive signals. **MStein** [188] models

 doi:10.20944/preprints202601.2218.v1

sequences as Gaussian distributions to capture view uncertainty, using Wasserstein discrepancy to minimize distance between positive pairs, improving tolerance to non-overlapping perturbed distributions. **LMA4Rec** [189] adopts Learnable Bernoulli Dropout (LBD) to generate positive views, training the model to ignore perturbation-prone signals. The dropout rates are learned via gradient-based optimization, adaptively masking less critical neurons to preserve semantic consistency.

Discussion and Assessment (detailed in Table 4). SSL-based RSRs mitigate behavioral randomness via contrastive view construction, effectively addressing partial mismatch and enhancing training-phase robustness. However, they overlook contextual influences, malicious manipulations, complete mismatch, inference-phase robustness, and motivational transformations. Practically, they are generally model-agnostic, though standalone architectures (e.g., ContrastVAE, RAP) lack flexibility; most rely solely on user-item interaction data, with a subset (e.g., FCLRec, BHGCL) demanding auxiliary features. Lightweight feature/sequence-level methods scale well, but those involving complex clustering or hypergraph operations incur overhead in large-scale scenarios, and formal robustness guarantees are absent.

5.3.5. Learning-Centric RSRs Based on Causal Learning

Causal Learning enhances SRS robustness by disentangling genuine causal relationships between user behaviors and preferences from confounders (perturbations induced by extrinsic motivations). It leverages causal inference tools to model the data-generation process, capturing stable preference signals invariant to confounders. Existing Causal Learning-based RSRs can be categorized into two subgroups: *Counterfactual Sequence Intervention* and *Structural Causal Modeling*.

Subgroup 1: Counterfactual Sequence Intervention adopts counterfactual interventions to identify 'indispensable items' (genuine preferences) and 'dispensable items' (perturbations), thereby generating semantically meaningful counterfactual sequences to expand training data beyond observational instances. Specifically, **CauseRec** [44] defines an importance score to distinguish indispensable and dispensable items:

$$a_i = \mathrm{rel}(\mathbf{v}_i, \mathbf{v}_t^u), \forall v_i \in \mathbf{x}_t^u, \tag{30}$$

where function $\mathrm{rel}(\cdot,\cdot)$ measures the relevance between two items. Counterfactual input sequences $\tilde{\mathbf{x}}_t^{u,+}$ and $\tilde{\mathbf{x}}_t^{u,-}$ are generated by replacing dispensable items (low a_i) or indispensable items (high a_i):

$$\tilde{\mathbf{x}}_t^{u,+} = \mathrm{replace}(\mathbf{x}_t^u, \mathcal{C}_{\mathrm{disp}}) \quad \text{and} \quad \tilde{\mathbf{x}}_t^{u,-} = \mathrm{replace}(\mathbf{x}_t^u, \mathcal{C}_{\mathrm{indisp}}), \tag{31}$$

where $\mathcal{C}_{\mathrm{disp}}$ and $\mathcal{C}_{\mathrm{indisp}}$ denote sets of dispensable and indispensable items, respectively. The contrastive loss pushes the model to ignore dispensable perturbations and focus on indispensable preferences:

$$\mathcal{L}_{\mathrm{cf}} = \sum_{\langle \mathbf{x}_t^u, v_t^u \rangle \in \mathcal{D}_{\mathrm{train}}} \max(0, \mathrm{dis}(\mathbf{x}_t^u, \tilde{\mathbf{x}}_t^{u,+}) - \mathrm{dis}(\mathbf{x}_t^u, \tilde{\mathbf{x}}_t^{u,-}) + \delta), \tag{32}$$

where $\mathrm{dis}(\cdot,\cdot)$ is a distance metric and δ is a margin. **CASR** [190,191] generates counterfactual instances via random/frequency/RL-based replacement, filtering low-quality instances with a confidence threshold. **PACIFIC** [192] generates counterfactual sequences via learnable perturbations, ensuring small but target-altering perturbations to learn reliable causal relationships.

Subgroup 2: Structural Causal Modeling constructs explicit causal graphs to model relationships between variables(user u, input sequence $\mathbf{x}_t^u$, preference $\mathbf{p}_t^u$, confounder set $\mathcal{C}_{\mathrm{pertb}}$, and target v_t^u) and uses causal calculus (e.g., backdoor adjustment, do-calculus) to block spurious paths induced by perturbations. For a confounder $\mathbf{C}_t^u$ between $\mathbf{x}_t^u$ and v_t^u, the causal effect is estimated via backdoor adjustment:

$$P(v_t^u | \mathrm{do}(\mathbf{x}_t^u)) = \sum_{c \in \mathcal{C}_{\mathrm{pertb}}} P(v_t^u | \mathbf{x}_t^u, c) P(c). \tag{33}$$

In particular, **CoDeR** [193,194] constructs a causal graph with demand drift as a confounder, using backdoor adjustment and stability/deviation metrics to filter noisy shifts induced by perturbations. **CSRec** [195] disentangles perturbations from system influence by distinguishing observational data (natural user behavior) and interventional data (system-driven interactions). Its causal graph models relationships between system recommendations, user preferences, and decisions, with the causal effect estimated via do-calculus. **DePOI** [196] mitigates perturbations from popularity and geographical

 doi:10.20944/preprints202601.2218.v1

bias by disentangling the input graph into causal and bias subgraphs. An adaptive edge mask generator assigns edges to either subgraph, and dual regularization (causal-bias disentanglement and transition-geography disentanglement) ensures semantic independence between representations.

Discussion and Assessment (detailed in Table 4). Causal Learning-based RSRs handle behavioral randomness and contextual influences (as confounders) via counterfactual operations, excelling at partial mismatch and enhancing training-phase robustness. However, they neglect malicious manipulations, complete mismatch, inference-phase perturbations, and motivational transformations. Practically, some variants (CauseRec, CASR, CSRec) are model-agnostic, with CoDeR and DePOI demanding auxiliary data. Heuristic methods are efficient, but learning-based samplers incur significant overhead, and formal robustness guarantees are absent.

Table 4. Evaluation of Learning-centric RSRs (§ 5.3). ○, △, and × indicate a property is fully satisfied, partially satisfied, and unsatisfied, respectively.

Category		Method	P1 Multi-cause Robustness	P2 Dual-manifestation Robustness	P3 Dual-phase Robustness	P4 Motivation Transformation Awareness	P5 Generality	P6 Data Accessibility	P7 Scalability	P8 Theoretical Grounding
Adversarial Training (§ 5.3.1)	Gradient-based AT	AdvTrain [149]	△	△	△	×	○	○	×	×
		PamaCF [151]	△	△	△	×	○	○	×	○
		VAT [152]	△	△	△	×	○	○	×	×
		MVS [150]	△	△	△	×	○	○	×	×
		Qian et al. [153]	△	△	△	×	○	○	×	×
	Sequence-based AT	AdvGraph [154]	△	△	△	×	○	○	×	×
		DARTS [155]	△	△	△	×	○	○	×	×
Reinforcement Learning (§ 5.3.2)	Sequence Construction	HRL [156]	△	△	△	×	×	○	×	×
		SAR [159]	△	△	△	×	×	○	×	×
		RLNF [160]	△	△	△	×	×	○	×	×
		HRL4Ba [157]	△	△	△	×	×	○	×	×
		MHRR [158]	△	△	△	×	×	×	×	×
	Feature Selection	KERL [161]	△	△	△	×	×	×	×	×
		MARIS [162]	△	△	△	×	×	×	×	×
Distributionally Robust Optimization (§ 5.3.3)	Partition-based DRO	S-DRO [164]	△	△	△	○	○	○	○	△
		IDEA [165]	△	△	△	○	○	○	○	△
	Distribution-based DRO	DROS [166]	△	△	△	○	○	○	○	△
		RSR [167]	△	△	△	○	○	○	○	△
	Model-based DRO	DePoD [168]	△	△	△	○	○	○	×	△
		E-NSDE [169]	△	△	△	○	○	○	×	△
		IDURL [170]	△	△	△	○	○	×	×	△
Self-supervised Learning (§ 5.3.4)	Feature-level SSL	DCRec [171]	△	△	△	×	×	○	△	×
		FCLRec [172]	△	△	△	×	×	×	△	×
		SGCL [173]	△	△	△	×	×	○	△	×
		BHGCL [174]	△	△	△	×	×	×	△	×
	Sequence-level SSL	RAP [175]	△	△	△	×	×	○	△	×
		CLEA [176]	△	△	△	×	×	○	△	×
		ICL [177]	△	△	△	×	×	○	×	×
		BNCL [178]	△	△	△	×	×	○	△	×
		CL4Rec [?]	△	△	△	×	×	○	○	×
		ICSRec [180]	△	△	△	×	×	○	○	×
		IOCLRec [181]	△	△	△	×	×	○	△	×
		AsarRec [182]	△	△	△	×	○	○	○	×
	Model-level SSL	DuoRec [183]	△	△	△	×	○	○	△	×
		SHT [185]	△	△	△	×	○	○	×	×
		ContrastVAE [184]	△	△	△	×	×	○	△	×
		AdaptSSR [186]	△	△	△	×	×	○	△	×
		MCLRec [187]	△	△	△	×	×	○	△	×
		MStein [188]	△	△	△	×	×	○	△	×
		LMA4Rec [189]	△	△	△	×	×	○	△	×
Causal Learning (§ 5.3.5)	Counterfactual Sequence Intervention	CauseRec [44]	△	△	△	×	○	○	○	×
		CASR [190]	△	△	△	×	○	○	○	×
		PACIFIC [192]	△	△	△	×	×	○	△	×
	Structural Causal Modeling	CoDeR [193]	△	△	△	×	×	○	×	×
		CSRec [195]	△	△	△	×	×	○	×	×
		DePOI [196]	△	△	△	×	×	×	×	×
Curriculum Learning (§ 5.3.6)	Static Hardness Measurement	GNNO [197]	△	△	△	×	○	○	○	×
		Diff4Rec [144]	△	△	△	×	○	○	△	×
		MELT [198]	△	△	△	×	○	○	○	×
	Dynamic Hardness Measurement	CCI [200]	△	△	△	×	○	×	○	×
		HSD [199]	△	△	△	×	○	○	○	×
		CDR [201]	△	△	△	×	○	○	○	×
		EXHANS [202]	△	△	△	×	○	○	○	×

5.3.6. Learning-Centric RSRs Based on Curriculum Learning

Curriculum Learning enhances SRS robustness by organizing training from easy (reliable) to hard (potentially unreliable) instances, mimicking human educational principles. It allows the model to learn stable sequential patterns before exposure to ambiguous/unreliable instances, avoiding early-stage misleading signals. The core of Curriculum Learning is the measurement of instance hardness, based on which existing methods can be categorized into *Static Hardness Measurement* and *Dynamic Hardness Measurement*.

 doi:10.20944/preprints202601.2218.v1

Subgroup 1: Static Hardness Measurement quantifies instance hardness using intrinsic structural/distributional properties of interaction data, with fixed scores throughout training. Specifically, **GNNO** [197] defines hardness via Jaccard similarity of neighborhood overlap between target and candidate negative items—high overlap indicates false negatives (hard instances). **Diff4Rec** [144] measures hardness via cosine similarity between original sequences and diffusion-augmented sequences—low similarity indicates augmented sequences are perturbation-prone (hard) instances. **MELT** [198] quantifies hardness via sequence length—shorter sequences (limited stable signals) are deemed harder instances.

Subgroup 2: Dynamic Hardness Measurement calculate instance hardness dynamically during training using real-time model feedback (e.g., prediction error, loss, consistency) Specifically, **HSD** [199] Defines hardness via user-level (target deviation from long-term preferences) and sequence-level (target deviation from local context) inconsistency—higher inconsistency indicates harder instances (potentially unreliable instances). **CCL** [200] measures hardness via Jensen-Shannon (JS) divergence between predicted and ground-truth user attribute distributions—higher divergence indicates harder instances. **CDR** [201] and **EXHANS** [202] define hardness as the absolute difference between predicted interaction probability and ground-truth labels—larger errors indicate harder instances.

Discussion and Assessment (detailed in Table 4) Curriculum Learning-based RSRs address behavioral randomness via hardness-aware scheduling, excelling at complete mismatch and enhancing training-phase robustness by prioritizing consistent instances. However, they overlook contextual influences, malicious manipulations, partial mismatch, inference-phase robustness, and motivational transformations. Practically, they are model-agnostic, with some methods (e.g., CCL, EXHANS) demanding auxiliary features. Scalability is constrained by offline pre-processing overhead for static methods and real-time hardness calculation cost for dynamic ones, and formal robustness guarantees are absent.

5.4. Inference-Centric RSRs

Inference-centric RSRs enhance robustness at the final prediction stage of sequential recommendation. In contrast to architecture-, data-, and learning-centric approaches that focus on preventing the model from acquiring erroneous sequential patterns during training, inference-centric RSRs explicitly embrace the presence of perturbations in real-time user sequences. Their objective is to generate balanced, comprehensive, and motivation-aligned recommendation lists that reflect users' intrinsic preferences rather than being biased by localized perturbations in the input. Two major methodological paradigms dominate this space: *Recommendation Calibration* and *Multi-interest Disentanglement*. Figure 9 illustrates the taxonomy and evolution of these approaches.

5.4.1. Inference-Centric RSRs Based on Recommendation Calibration

Recommendation calibration improves inference-phase robustness by aligning the attribute distribution of the recommended items (e.g., genres, categories) with that of the user's historical interactions. By enforcing distributional consistency, calibration mitigates the skew induced by perturbations and ensures that recommended items fully cover users' underlying motivations. Existing calibration-based RSRs fall into *Post-processing Calibration* and *End-to-End Calibration*.

 doi:10.20944/preprints202601.2218.v1

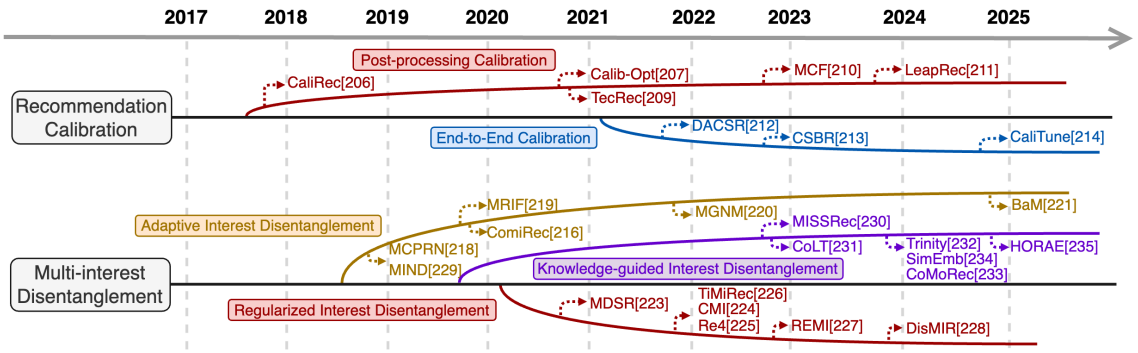

Figure 9. Development trajectory of Inference-centric RSRs.

Subgroup 1: Post-processing Calibration adjusts the recommendation list produced by a base SRS through a re-ranking step. Given an input sequence $\mathbf{x}_t^u$ and an initial recommendation list R_t^u, the attribute distributions of the history and the recommended list are computed as:

$$p_{\text{his}}(g|\mathbf{x}_t^u) = \frac{\sum_{v \in x_t^u} \mathbb{I}(g \in \text{attrib}(v))}{\sum_{g' \in \mathcal{G}_{\text{attr}}} \sum_{v \in \mathbf{x}_t^u} \mathbb{I}(g' \in \text{attrib}(v))}, \quad p_{\text{rec}}(g|R_t^u) = \frac{\sum_{v \in R_t^u} \mathbb{I}(g \in \text{attrib}(v))}{\sum_{g' \in \mathcal{G}_{\text{attr}}} \sum_{v \in R_t^u} \mathbb{I}(g' \in \text{attrib}(v))}, \tag{34}$$

where $\mathcal{G}_{\text{attr}}$ is the set of item attributes (e.g., genres), $g \in \mathcal{G}_{\text{attr}}$ is an attribute, $\mathbb{I}(\cdot)$ is the indicator function, and $\text{attrib}(v)$ returns the attribute set for item v. A calibrated list is obtained by optimizing:

$$R_t^{u*} = \arg \max_{R_t^u \subseteq \mathcal{V}, |R_t^u| = K} \left[(1-\lambda) \cdot \text{rel}(R_t^u) - \lambda \cdot \text{div}(p_{\text{his}}, p_{\text{rec}}) \right], \tag{35}$$

where $\text{rel}(R_t^u)$ is the relevance of the list (e.g., sum of predicted scores) and, $\text{div}(p_{\text{his}}, p_{\text{rec}})$ measures the divergence between p_{his} and p_{rec}, and λ controls the trade-off between relevance and calibration. This approach calibrates the list distribution to match historical patterns, reducing skewness towards perturbations. Specifically **CaliRec** [203] uses Kullback-Leibler (KL) divergence to reliaze $\textbf{div}(\cdot)$, and solves the re-ranking objective (Equation 35) with greedy search. **Calib-Opt** [204] measures distribution divergence via total variation [205], and formulates calibration as a constrained optimization problem, which provides linearity and computational tractability. **TecRec** [206] predicts evolving attribute distributions using LSTM-based models, performing calibration by dynamically adapting to preference shifts. **MCF** [207] models calibration as a minimum-cost flow problem, assigning higher costs to items that skew the attribute distribution, thereby pruning perturbation-induced outliers with polynomial-time optimality. **LeapRec** [208] proposes a two-phase framework: calibration-disentangled learning (separating relevance and calibration signals) and relevance-prioritized re-ranking (ensuring top-ranked items remain relevant while calibrating the overall list).

Subgroup 2: End-to-End Calibration integrates calibration objectives directly into the training process of SRSs, eliminating the additional computational overhead of post-processing. By jointly optimizing recommendation accuracy and calibration via a hybrid loss, the model learns to generate calibrated recommendations during inference. Let $\mathbf{r}_t^u$ be the score vector of all items predicted by the SRS for input sequence $\mathbf{x}_t^u$. The predicted attribute distribution of the recommendation list is:

$$\hat{p}_{\text{rec}}(g|x_t^u) = \sum_{v \in \mathcal{V}} \text{softmax}(\mathbf{r}_t^u)_v \cdot \mathbb{I}(g \in \text{attributes}(v)). \tag{36}$$

The total training loss unifies standard recommendation loss and calibration loss:

$$\mathcal{L}_{\text{calib}} = \sum_{\langle \mathbf{x}_t^u, v_t^u \rangle \in \mathcal{D}_{train}} [\phi(f(\mathbf{x}_t^u, v_t^u)) + \lambda \cdot \text{div}(\hat{p}_{\text{rec}}(g|x_t^u), p_{\text{his}}(g|\mathbf{x}_t^u))], \tag{37}$$

where $\phi(\cdot)$ is the base recommendation loss (e.g., cross-entropy, BPR). In particular, **DACSR** [209] designs a decoupled-aggregated architecture with separate encoders for accuracy and calibration, fused via an extractor network, avoiding the 'seesaw effect' of joint optimization. **CSBR** [210] incorporates a calibration module to match the long-tail item ratio in recommendations with the historical ratio. It uses subset-specific encoders for imbalanced data, preventing perturbations from overwhelming the model with head-item dominated signals. **CaliTune** [211] fine-tunes pre-trained SRSs with a

 doi:10.20944/preprints202601.2218.v1

list-wise loss that jointly optimizes relevance and calibration, using dynamic candidate sampling to filter perturbations that fall out of the candidate pool during fine-tuning.

Discussion and Assessment (detailed in Table 5). Calibration-based RSRs handle behavioral randomness via distributional alignment, effectively addressing partial mismatch and enhancing inference-phase robustness. Yet, they overlook contextual influences, malicious manipulations, complete mismatch, training-phase robustness, and motivational transformations. Practically, post-processing calibration is model-agnostic but all variants require item attributes beyond raw interaction data. Scalability varies: post-processing methods incur additional inference costs, while end-to-end approaches face higher training overhead, and formal robustness guarantees are absent.

5.4.2. Inference-Centric RSRs Based on Multi-Interest Disentanglement

Perturbed interactions disproportionately distort single-vector user representations in traditional SRSs, causing the final output to focus on noisy or transient signals. Multi-interest disentanglement addresses this by decomposing user preferences into multiple independent interest vectors, each representing a coherent facet of user motivation. The resulting representation reduces perturbation impact and yields more comprehensive and robust inference. Existing Multi-interest Disentanglement-based RSRs can be categorized into three subgroups: *Adaptive Interest Disentanglement*, *Regularized Interest Disentanglement*, and *Adaptive Interest Disentanglement*.

Subgroup 1: Adaptive Interest Disentanglement dynamically clusters user interactions into multiple interest representations via unsupervised, data-driven mechanisms (e.g., attention, dynamic routing) without external guidance. These methods adaptively weight sequence items based on their relevance to each interest, isolating perturbations by down-weighting irrelevant items within each cluster. The core formulation for learning disentangled interest vectors $\mathbf{z}_{t,m}^{u}$ (the m-th interest of user u at time t) is: $\mathbf{z}_{t,m}^{u} = \sum_{v \in \mathbf{x}_t^u} \alpha_{m,v} \cdot \mathbf{v}$, where $\alpha_{m,v}$ is the adaptive weight of item v for the m-th interest (learned via attention/routing). In particular, **MIND** [212] and **ComiRec** [213] use dynamic routing [214] to generate multiple interest vectors, assigning perturbations to less dominant interests to reduce their impact. **MCPRN** [215] proposes a mixture-channel structure with purpose-specific routing, filtering cross-purpose perturbations via soft assignment to dedicated interest channels. **MRIF** [216] leverages multi-resolution aggregation and attention fusion to model interests at different temporal scales, smoothing out high-frequency perturbations. **MGNM** [217] integrates graph convolution and sequential capsule networks to capture hierarchical interest relationships, using graph smoothing to mitigate perturbations. **BaM** [218] proposes a soft-selection training scheme that balances interest learning via probabilistic sampling, preventing perturbations from overshadowing genuine preferences.

Subgroup 2: Regularized Interest Disentanglement enhances disentanglement and avoids interest collapse (i.e., redundant interest vectors) by incorporating auxiliary regularization terms into the training objective. Common regularizers include diversity loss (encouraging orthogonality between interest vectors) and consistency loss (aligning interest assignments with sequential coherence). Specifically, **IDSR** [219] and **MDSR** [220] use implicit interest mining modules with diversity-promoting losses to disentangle interests, isolating perturbations into specific vectors to prevent global corruption. **CMI** [221] applies contrastive learning to encorce consistency across interest representations disentangled from augmented sequence views, making interests robust to perturbations. **Re4** [222] introduces backward flow mechanisms (re-contrast, re-attend, re-construct) to regularize interest learning, aligning attention weights with recommendation goals to reduce perturbation effects. **TiMiRec** [223] introduces target interest distillation, dynamically aggregating multi-interest vectors during inference via a target-interest predictor. The predictor is supervised by a distillation loss using target-item similarity as soft labels, ensuring interests align with genuine intent rather than perturbations. **REMI** [224] employs interest-aware hard negative mining and routing regularization to balance interest training, minimizing the influence of easy negatives induced by perturbations. **DisMIR** [225] uses spectral clustering-inspired regularization to prevent interest collapse, leveraging global item co-occurrence patterns to enhance robustness against sparse perturbations.

 doi:10.20944/preprints202601.2218.v1

Table 5. Evaluation of Inference-centric RSRs (§ 5.4). ○, △, and × indicate a property is fully satisfied, partially satisfied, and unsatisfied, respectively.

Category		Method	P1 Multi-cause Robustness	P2 Dual-manifestation Robustness	P3 Dual-phase Robustness	P4 Motivation Transformation Awareness	P5 Generality	P6 Data Accessibility	P7 Scalability	P8 Theoretical Grounding
Recommendation Calibration (§ 5.4.1)	Post-processing Calibration	CaliRec [203]	△	△	△	×	○	×	×	×
		Calib-Opt [204]	△	△	△	×	○	×	×	×
		TecRec [206]	△	△	△	×	○	×	×	×
		MCF [207]	△	△	△	×	○	×	×	×
		LeapRec [208]	△	△	△	×	○	×	×	×
	End-to-End Calibration	DACSR [209]	△	△	△	×	×	×	△	×
		CSBR [210]	△	△	△	×	×	×	△	×
		CaliTune [211]	△	△	△	×	×	×	△	×
Multi-interest Disentanglement (§ 5.4.2)	Adaptive Interest Disentanglement	MIND [?]	△	△	△	×	×	×	×	×
		MCPRN [215]	△	△	△	×	×	×	×	×
		MRIF [216]	△	△	△	×	×	○	×	×
		MGNM [217]	△	△	△	×	×	○	△	×
		BaM [218]	△	△	△	×	×	○	○	×
		ComiRec [213]	△	△	△	×	×	×	×	×
	Regularized Interest Disentanglement	IDSR [219]	△	△	△	×	×	×	×	×
		MDSR [220]	△	△	△	×	×	×	×	×
		CMI [221]	△	△	△	×	×	○	△	×
		Re4 [222]	△	△	△	×	×	○	△	×
		TiMiRec [223]	△	△	△	×	×	○	×	×
		REMI [224]	△	△	△	×	×	○	×	×
		DisMIR [225]	△	△	△	×	×	○	×	×
	Knowledge-guided Interest Disentanglement	MISSRec [226]	△	△	△	×	×	×	×	×
		CoLT [227]	△	△	△	×	×	○	△	×
		Trinity [228]	△	△	△	×	×	×	×	×
		CoMoRec [229]	△	△	△	×	×	×	×	×
		SimEmb [230]	△	△	△	×	×	×	×	×
		HORAE [231]	△	△	△	×	×	○	×	×

Subgroup 3: Knowledge-guided Interest Disentanglement integrates external knowledge (e.g., knowledge graphs, multi-modal data, item attributes) to provide semantic guidance for interest separation, reducing reliance on sparse interaction data and enhancing robustness to perturbations (especially in cold-start scenarios). In particular, **MISSRec** [226] proposes a multi-modal pre-training framework that generates ID-agnostic item embeddings from text and image features. It clusters multi-modal tokens into interest prototypes, reducing vulnerability to perturbations in sparse interaction data. **CoLT** [227] addresses the long-tail problem by leveraging co-occurrence relationships between head and tail items. It constructs a co-occurrence adjacency matrix to weight head-item embeddings, injecting semantic information into tail-item representations and enhancing their resistance to perturbations. **Trinity** [228] builds a global item clustering system based on co-occurrence patterns, mapping user long-term sequences to interest histograms. Three dedicated retrievers (Trinity-M for multi-interest, Trinity-LT for long-tail interest, Trinity-L for long-term interest) use statistical aggregation to mitigate transient perturbations. **CoMoRec** [229] integrates conversational context, temporal knowledge graphs, and item reviews to model multi-aspect interests, providing a dense semantic structure that suppresses perturbations. **SimEmb** [230] simulates item attributes from interaction data via co-occurrence matrices, replacing ID-based embeddings with attribute-weighted sums. This sharpens item clusters and enhances robustness to attribute scarcity and perturbations. **HORAE** [231] incorporates fine-grained temporal dynamics into multi-interest pre-training, modeling relative positions, adjacent time intervals, and target time gaps via rotary position embeddings and time-binning. Sequential interest refinement captures interest drift, enhancing robustness against time-varying perturbations.

Discussion and Assessment (detailed in Table 5). Multi-interest Disentanglement-based RSRs handle behavioral randomness via interest isolation, excelling in inference-phase robustness and partial mismatch by covering multiple user motivations. Yet, they overlook adversarial manipulations, complete mismatch, training-phase robustness, and motivational transformations. They require significant architectural adaptation with knowledge-guided variants demanding auxiliary data; scalability is constrained by complex routing or knowledge integration. Formal theoretical guarantees for robustness are absent.

6. Evaluation Metrics and Benchmarks of RSRs

This section focuses on the experimental evaluation of RSRs, covering two core aspects: robustness-oriented evaluation metrics and widely used benchmark datasets.

 doi:10.20944/preprints202601.2218.v1

6.1. Evaluation Metrics of RSRs

Evaluating RSRs requires metrics that align with their dual-phase robustness objectives (Section 2.3). Unlike conventional recommendation metrics that merely prioritize accuracy, RSR metrics must quantify training-phase robustness (ability to resist perturbation-induced erroneous patterns) and inference-phase robustness (ability to generate preference-aligned recommendations despite input perturbations). This section reorganizes RSR metrics based on their core role in robustness assessment, with clear alignment to the dual-phase robustness. Table 6 consolidates the definitions and usage of these metrics across the RSR literature.

Table 6. Evaluation Metrics of RSRs.

Category		Metric	Definition	Requirements	Representative Publications
Training-phase Robustness Metrics	/	Relative Improvements (RI)	$\frac{M(f_{\text{robust}})-M(f_{\text{origin}})}{M(f_{\text{origin}})}$	Recommendation Lists	[14,15,25,137?,138]
Inference-phase Robustness Metrics	Diversity Metrics	Intra-List Distance@K (ILD@K)	$\frac{1}{K(K-1)}\sum_{v_i\in R_t^u}\sum_{v_j\in R_t^u, i\neq j}\text{dis}(v_i,v_j)$	Recommendation Lists Item Embeddings	[129,219,220]
		Diversity@K	$-\sum_{i=1}^{K} P_i \log P_i$	Recommendation Lists Item Categories	[215]
	Calibration Metrics	KL Calibration (KLC)	$\sum_c p(c\|\mathbf{x}_t^u)\log\frac{p(c\|\mathbf{x}_t^u)}{q(c\|R_t^u)}$	Recommendation Lists Item Categories	[203,204,207,208]
		Total Variation (TV)	$\frac{1}{2}\sum_c \left\|p(c\|\mathbf{x}_t^u)-q(c\|R_t^u)\right\|$	Recommendation Lists Item Categories	[204]

6.1.1. Training-Phase Robustness Metrics

Training-phase robustness metrics measure the model's capacity to learn authentic sequential patterns without being misled by unreliable instances. Researchers typically adopt Relative Improvement (**RI**) to quantify the model's resilience to perturbations, calculated by comparing accuracy metrics before and after integrating robust mechanisms [14,15,25]:

$$\text{RI} = \frac{M(f_{\text{robust}}) - M(f_{\text{origin}})}{M(f_{\text{origin}})}, \tag{38}$$

where f_{robust} denotes the recommender that integrates robustness mechanisms based on the original recommender f_{origin}, and $M(\cdot)$ represents standard accuracy metrics, such as Precision [232], Recall [232], Area Under the ROC Curve (AUC) [233], Mean Reciprocal Rank (MRR) [233], Normalized Discounted Cumulative Gain (NDCG) [234], etc. A positive RI value indicates that the robustness mechanism mitigates the adverse impacts of unreliable instances during training.

6.1.2. Inference-Phase Robustness Metrics

Inference-phase robustness metrics assess whether recommendations align with users' intrinsic preferences despite input perturbations. These metrics are categorized into two groups: diversity-based metrics (quantifying perturbation-induced skewness) and calibration-based metrics (evaluating the alignment between recommendation distributions and historical preference distributions).

Diversity-based Metrics are initially designed to quantify the diversity of recommendation lists [235]. In the context of RSR, these metrics are adopted to evaluate whether perturbations skew recommendations toward homogeneous items. Specifically, Intra-List Distance@K (**ILD@K**) measures average dissimilarity between pairs of items in the top-K recommendation list R_t^u:

$$\text{ILD@}K = \frac{1}{K(K-1)} \sum_{v_i \in R_t^u} \sum_{v_j \in R_t^u, i \neq j} \text{dis}(v_i, v_j). \tag{39}$$

A high ILD@K value indicates that perturbations do not skew the model to recommend homogeneous items, reflecting preserved diversity. **Diversity@K** uses entropy to measure category diversity in the top-K recommendation list:

$$\text{Diversity@}K = -\sum_{i=1}^{K} P_i \log P_i, \tag{40}$$

where P_i is the proportion of items in the list belonging to item v_i's category. A high Diversity@K value indicates that the model does not over-concentrate on a single category due to perturbations.

 doi:10.20944/preprints202601.2218.v1

Calibration Metrics directly measure the alignment between recommendation distributions and users' historical preference distributions. Specifically, KL Calibration (**KLC**) uses KL divergence to compare historical and recommended item category distributions:

$$KLC = \sum_c p(c|\mathbf{x}_t^u) \log \frac{p(c|\mathbf{x}_t^u)}{q(c|R_t^u)}, \tag{41}$$

where c denotes a specific item category (e.g., 'action' for movies), $p(c|\mathbf{x}_t^u)$ and $\mathbf{x}_t^u$, $q(c|R_t^u)$ are the proportions of category c in the input sequence $\mathbf{x}_t^u$ and recommendation list R_t^u, respectively. A low C_{KL} value indicates recommendations are well-calibrated to the user's historical preferences, reflecting minimal distortion from perturbations. Total Variation (**TV**) provides a linear-scaled measure of calibration:

$$\text{TV} = \frac{1}{2} \sum_c |p(c|\mathbf{x}_t^u) - q(c|R_t^u)|, \tag{42}$$

where $\frac{1}{2}$ is a normalization factor, ensuring the value ranges between 0 and 1. A lower KLC value indicates recommendations are more consistent with historical preference distributions.

6.1.3. Discussion of RSR Metrics

Despite the widespread use of the aforementioned metrics in RSR evaluation, existing practices still face three key limitations that hinder comprehensive and accurate assessment of model robustness:

(i) *Incomplete dual-phase assessment*. Most existing RSRs rely solely on training-phase or inference-phase metrics in isolation, ignoring the complementarity of the two phases. This one-sided assessment fails to capture the holistic robustness of RSRs, as a model may perform well in training but fail to generalize to perturbed input sequences during inference (and vice versa).

(ii) *Conflation of perturbation skewness and preference concentration*. Current diversity-based metrics fail to disentangle perturbation-induced skewness and users' inherently focused preferences, both of which lead to low diversity values, resulting in misjudgments of model robustness. However, addressing such conflation fundamentally relies on explicit motivation annotations, which are absent in existing datasets (detailed in Section 7).

(iii) *Over reliance on high-quality category attributes*. Most Inference-phase robustness metrics (except ILD@K) depend on predefined item category attributes, which are often manually annotated, heuristically derived, or incomplete in real-world scenarios. The lack of standardized, high-quality category information limits the generalizability of these metrics across different recommendation domains.

6.2. Benchmarks for Evaluating RSRs

RSR research relies heavily on rich benchmarks to validate methodological effectiveness, simulate real-world perturbation scenarios, and ensure result reproducibility. This section summarizes the key statistics of representative benchmarks used in existing RSR literature and discusses the limitations of existing usages.

6.2.1. Statistics of Representative Benchmarks

Through a systematic statistical analysis of 87 datasets cited in our surveyed RSR-related publications, we identify 27 representative benchmarks with no fewer than 4 citations (as illustrated in Figure 10), which collectively account for 81.58% of the total citations in the field. These datasets span diverse application domains, including E-commerce, Movie, Game, Music, News, and Location-Based Social Networks (LBSN). They exhibit significant variations in scale (ranging from hundreds of thousands to millions of users/items/interactions) and auxiliary feature availability (e.g., user demographics, item attributes, social networks, review texts), as comprehensively summarized in Table 7.

 doi:10.20944/preprints202601.2218.v1

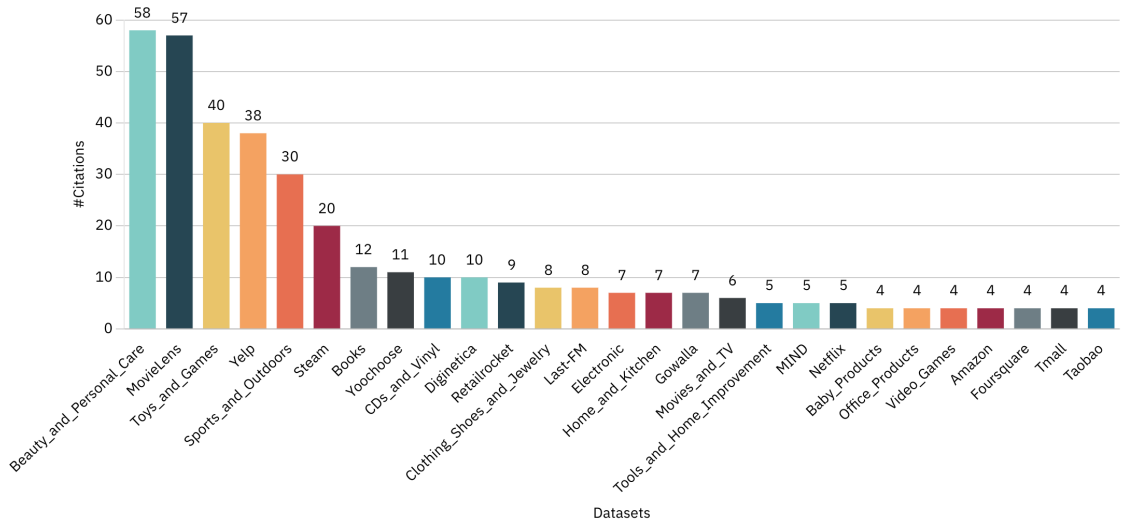

Figure 10. Number of citations of representative benckmarks in the RSR literature.

Table 7. Evaluation of Architecture-centric RSRs (§ 5.1).

Dataset	Domain	#Users	#Items	#Interactions	Features beyond User-item Interaction
Beauty_and_Personal_Care	E-commerce	11.3M	1.0M	23.9M	User-item Review, Item Auxiliary Info
MovieLens	Movie	200.9K	87.6K	32.0M	User Auxiliary Info, Item Auxiliary Info
Toys_and_Games	E-commerce	8.1M	890.7K	16.3M	User-item Interaction, Item Auxiliary Info
Yelp	E-commerce	30.4K	20.0K	316.3K	Item Auxiliary Info
Sports_and_Outdoors	E-commerce	10.3M	1.6M	19.6M	User-item Review, Item Auxiliary Info
Steam	Game	2.6M	15.5K	78M	User-item Review, Item Auxiliary Info
Books	E-commerce	10.3M	4.4M	29.5M	User-item Review, Item Auxiliary Info
Yoochoose	E-commerce	115.6K	24.1K	1.8M	Item Auxiliary Info
CDs_and_Vinyl	E-commerce	1.8M	701.7K	4.8M	User-item Review, Item Auxiliary Info
Diginetica	E-commerce	780.3K	43.1K	982.9K	/
Retailrocket	E-commerce	1.4M	417.1K	2.8M	Item Auxiliary Info
Clothing_Shoes_and_Jewelry	E-commerce	22.6M	7.2M	66.0M	User-item Review, Item Auxiliary Info
Last-FM	Music	23.6K	48.1K	3.0M	Item Auxiliary Info, User-User Social Network
Electronic	E-commerce	18.3M	1.6M	43.9M	User-item Review, Item Auxiliary Info
Home_and_Kitchen	E-commerce	23.2M	3.7M	67.4M	User-item Review, Item Auxiliary Info
Gowalla	LBSN	196.6K	1.3M	6.4M	User-User Social Network
Movies_and_TV	E-commerce	6.5M	747.8K	17.3M	User-item Review, Item Auxiliary Info
Tools_and_Home_Improvement	E-commerce	12.2M	1.5M	27.0M	User-item Review, Item Auxiliary Info
MIND	News	161.0K	317.1K	24.2M	Item Auxiliary Info, User-item Non-click Event
Netflix	Movie	463.4K	17.8M	57.0M	/
Baby_Products	E-commerce	3.4M	217.7K	6.0M	User-item Review, Item Auxiliary Info
Office_Products	E-commerce	7.6M	710.4K	12.8M	User-item Review, Item Auxiliary Info
Video_Games	E-commerce	2.8M	137.2K	4.6M	User-item Review, Item Auxiliary Info
Amazon	E-commerce	6.2K	2.8K	587.4K	User-item Review, Item Auxiliary Info
Foursquare	LBSN	114.3K	3.8M	22.8M	User Auxiliary Info, Item Auxiliary Info, User-User SocialNetwork
Tmall	E-commerce	424.2K	1.1M	55.0M	User Auxiliary Info, Item Auxiliary Info
Taobao	E-commerce	988.0K	4.2M	100.2M	Item Auxiliary Info

6.2.2. Discussion of RSR Benchmarks

Despite the richness and diversity of existing benchmarks, their usage in RSR research suffers from four core limitations that severely hinder the reproducibility, comparability, and validity of experimental results:

(i) *Absence of ground-truth reliability annotations*. Current benchmarks lack labels to distinguish between reliable and unreliable instances. This limits supervised RSR training and weakens the validity of robustness evaluation (as detailed in Challenge 1, Section 4).

(ii) *Lack of dataset selection standards*. Disparate benchmark choices across studies hinder fair comparison. We advocate prioritizing high-impact, frequently cited datasets per domain: MovieLens (Movie), Beauty_and_Personal_Care (E-commerce), Last-FM (Music), Steam (Game), MIND (News), and Gowalla (LBSN).

(iii) *No unified pre-processing protocols*. Variable filtering thresholds for inactive users and cold items lead to inconsistent data distributions, which hinder direct cross-study comparison. We

 doi:10.20944/preprints202601.2218.v1

appeal to standardize thresholds to 5, which balances sparsity reduction and preservation of meaningful behavioral patterns [236].

(iv) *Unstandardized Negative Sampling*. To balance computational efficiency and evaluation feasibility in large-scale item catalogs, many RSR studies [13,23,80] adopt diverse negative sampling strategies (e.g. random/hard sampling) [237]. However, such practices also restrict cross-study comparison and lead to biased evaluation [238]. Given the advancements in GPU-accelerated computing that enable efficient full-item-set ranking, we advocate abandoning sampling-based negative item selection in RSR evaluation.

7. Open Issues and Future Directions

Despite substantial progress, the field of RSR remains emerging with unresolved challenges. This section outlines six critical future directions, each targeting a core gap in the current landscape.

Direction 1: Synthesizing Data with Explicit Motivation Annotations. As discussed in Sections 4 (Challenge 1), 6.1.3, and 6.2.2, a fundamental bottleneck in RSRs is the scarcity of large-scale, high-quality datasets annotated with interaction motivations (intrinsic/extrinsic). This hinders effective model training—where RSRs rely on imperfect proxies (e.g., heuristics, self-supervised signals) to distinguish genuine preferences from perturbations—and rigorous evaluation—where synthetic perturbation injection (e.g., random item replacement) fails to reflect real-world complexity, undermining robustness assessment validity. Therefore, future research could focus on (1) *LLM-based Data Synthesis*: leverage LLMs' semantic reasoning to annotate motivations for existing datasets. Design prompts integrating user history, item attributes, and context to estimate the likelihood of interactions being driven by intrinsic preferences versus extrinsic factors. (2) *Controlled User Studies*: conduct targeted user studies [12] to collect small-scale, high-quality datasets with self-reported or expert-inferred motivations. These could serve as benchmarks for validating automated annotations and evaluating fine-grained RSR performance. (3) *Causal Data Generation*: extend counterfactual data augmentation frameworks [190] to label intervened items as perturbations, creating controlled environments for studying specific types of unreliable instances.

Direction 2: Theoretically-Grounded RSRs. Most existing RSRs are empirically driven, lacking solid theoretical foundations to guarantee robustness properties. Formal frameworks defining RSR robustness criteria (e.g., generalization bounds, perturbation stability, inference convergence) are scarce, leading to heuristic progress and unclear applicability boundaries. Possible solutions to theoretically sound RSRs encompass (1) *Algorithmic Stability Theory*: borrow tools from algorithmic stability theory [239] to quantify RSR sensitivity to training data perturbations. Derive generalization bounds that explicitly account for the proportion of unreliable instances. (2) *Causal Inference Frameworks*: formalize the data-generation process via structural causal models (SCMs) [240]. Define robustness as the model's ability to estimate the true causal effect of user history on next actions, decoupled from confounding extrinsic motivations.

Direction 3: Harmonizing Training- and Inference-phase Robustness. As analyzed in Section 2.3, RSRs face a 'precision-coverage dilemma' due to divergent phase requirements: training-phase robustness demands precise filtering of irrelevant items to avoid spurious patterns, while inference-phase robustness requires comprehensive coverage of user motivations for balanced recommendations. Most current methods prioritize one phase, leading to either incomplete inference coverage or corrupted training signals. Potential solutions include (1) *Multi-Objective and Adaptive Learning*: develop training paradigms that explicitly optimize for both objectives. This could involve adaptive loss functions that start with a focus on denoising (training-phase robustness) and gradually shift towards encouraging diversity and coverage (inference-phase robustness) as training progresses. (2) *Two-Stage Cascaded Architectures*: design separate but interconnected components: a first-stage model for reliability estimation and sequence correction (training-phase robustness), and a second-stage model for generating calibrated, comprehensive recommendations (inference-phase robustness). (3) *Adaptive*

 doi:10.20944/preprints202601.2218.v1

Inference Mechanisms: enable models to adjust denoising aggressiveness based on input sequence ambiguity—prioritizing precision for consistent sequences and coverage for ambiguous ones.

Direction 4: Unified RSRs for Multi-cause Unreliable Instances. As detailed in Section 3.1, real-world unreliable instances stem from coexisting causes (behavioral randomness, contextual influences, malicious manipulations), but most RSRs target only a subset of causes. Strategies effective for one cause (e.g., probabilistic modeling for randomness) may be ineffective or counterproductive for others (e.g., adversarial attacks), calling for unified frameworks. Viable approaches involve (1) *Mixture-of-expert (MoE) Perturbation Modeling*: design a gating network to classify unreliability causes for each instance, followed by specialized expert heads tailored to handle specific perturbation types [241]. (2) *Meta-learning for Cause Adaptation*: train RSRs on datasets with diverse extrinsic motivation mixes, with a meta-optimizer that adjusts model parameters to adapt to the cause distribution of input data, enabling generalization to unseen real-world cause combinations.

Direction 5: RSRs for Intrinsic-Extrinsic Motivation Transformation. As discussed in Section 4, user motivations are dynamic: extrinsic-driven interactions (e.g., friend recommendations) can evolve into intrinsic preferences, and vice versa. Current RSRs treat motivations as static, leading to two critical errors: premature discarding of emerging intrinsic signals, or persistent retention of outdated intrinsic interactions that now reflect extrinsic drivers. A promising solution could be *Reinforcement Learning with Long-Term Rewards*: frame the problem as an RL task where the agent (RSR) decides whether to 'trust' historical interactions. The reward function is tied to long-term user satisfaction, incentivizing the model to identify interactions with varying value considering motivational transformations.

Direction 6: Balancing Robustness, Personalization, and Scalability. As discussed in Section 4, RSRs face an inherent trilemma: aggressive robustness (e.g., strict filtering) degrades personalization for long-tail users/items; highly personalized models are perturbation-sensitive and computationally expensive; and scalable methods often compromise robustness or personalization. Achieving all three properties simultaneously remains unresolved. A possible strategy for addressing such a trilemma could be *Adaptive Resource Allocation*: develop RSRs that dynamically allocate computational resources based on user risk profiles and personalization needs. For example, apply fine-grained robustness mechanisms to high-risk users (e.g., new users with sparse histories) and lightweight methods to low-risk users. Prioritize long-tail items/users in robustness processing to preserve niche preferences.

8. Conclusion

This survey presents the first systematic and comprehensive review of the emerging field of Robust Sequential Recommenders (RSRs). We first analyzed unreliable instances in depth, clarifying their root causes, distinct manifestations, and multi-stakeholder adverse impacts on the recommendation ecosystem. Building on this foundation, we delineated the unique challenges of sequential robustness—challenges absent in non-sequential recommendation and general denoising tasks. To organize the expanding literature, we proposed a holistic lifecycle-based taxonomy, categorizing existing RSRs into four core paradigms (architecture-centric, data-centric, learning-centric, and inference-centric) and conducting systematic comparative analyses of representative methods via an eight-property assessment framework, highlighting their strengths and limitations. We also consolidated standardized evaluation metrics and benchmarks for RSR research, and identified open issues alongside promising future directions to advance the field.

Driven by the imperative for reliable and trustworthy sequential recommenders in real-world scenarios, RSR research is progressing rapidly. We anticipate this survey will serve as a foundational reference, offering researchers a clear grasp of the state-of-the-art and illuminating the path toward developing more robust, accurate, and efficient sequential recommenders.

References

1. Wang, S.; Cao, L.; Wang, Y.; Sheng, Q.Z.; Orgun, M.A.; Lian, D. A Survey on Session-based Recommender Systems. *ACM Computing Surveys (CSUR)* **2022**, *54*, 154:1–154:38.

 doi:10.20944/preprints202601.2218.v1

2. Ge, Y.; Liu, S.; Fu, Z.; Tan, J.; Li, Z.; Xu, S.; Li, Y.; Xian, Y.; Zhang, Y. A Survey on Trustworthy Recommender Systems. *ACM TORS* **2024**, *3*.
3. Wang, S.; Hu, L.; Wang, Y.; Cao, L.; Sheng, Q.Z.; Orgun, M. Sequential recommender systems: challenges, progress and prospects. In Proceedings of the IJCAI, 2019, pp. 6332–6338.
4. Fang, H.; Zhang, D.; Shu, Y.; Guo, G. Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations. *ACM TOIS* **2020**, *39*, 1–42.
5. McAuley, J.; Leskovec, J. Hidden factors and hidden topics: understanding rating dimensions with review text. In Proceedings of the RecSys, 2013, pp. 165–172.
6. Bao, Y.; Fang, H.; Zhang, J. TopicMF: Simultaneously Exploiting Ratings and Reviews for Recommendation. In Proceedings of the AAAI, 2014, Vol. 14, pp. 2–8.
7. Kim, D.; Park, C.; Oh, J.; Lee, S.; Yu, H. Convolutional matrix factorization for document context-aware recommendation. In Proceedings of the RecSys, 2016, pp. 233–240.
8. He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; Chua, T.S. Neural collaborative filtering. In Proceedings of the TheWebConf, 2017, pp. 173–182.
9. He, X.; Du, X.; Wang, X.; Tian, F.; Tang, J.; Chua, T.S. Outer product-based neural collaborative filtering. In Proceedings of the IJCAI, 2018.
10. Sarwar, B.; Karypis, G.; Konstan, J.; Riedl, J. Item-based collaborative filtering recommendation algorithms. In Proceedings of the TheWebConf, 2001, pp. 285–295.
11. Brühlmann, F.; Vollenwyder, B.; Opwis, K.; Mekler, E.D. Measuring the "Why" of Interaction: Development and Validation of the User Motivation Inventory (UMI). In Proceedings of the CHI, 2018, pp. 1–13.
12. Bennett, D.; Mekler, E.D. Beyond Intrinsic Motivation: The Role of Autonomous Motivation in User Experience. *ACM TOCHI* **2024**, *31*, 60:1–60:41.
13. Sun, Y.; Wang, B.; Sun, Z.; Yang, X. Does Every Data Instance Matter? Enhancing Sequential Recommendation by Eliminating Unreliable Data. In Proceedings of the IJCAI, 2021, pp. 1579–1585.
14. Sun, Y.; Yang, X.; Sun, Z.; Wang, B. BERD+: A Generic Sequential Recommendation Framework by Eliminating Unreliable Data with Item-and Attribute-level Signals. *ACM TOIS* **2023**, *42*, 1–33.
15. Sun, Y.; Wang, B.; Sun, Z.; Yang, X.; Wang, Y. Theoretically Guaranteed Bidirectional Data Rectification for Robust Sequential Recommendation. In Proceedings of the NeurIPS, 2023, pp. 2850–2876.
16. Ceci, L. TikTok: fake engagement prevented 2021-2024, 2024. https://www.statista.com/statistics/1318268/tiktok-fake-interactions-prevented/.
17. Community Standards Enforcement Report, Fake Accounts, 2024. https://transparency.meta.com/reports/community-standards-enforcement/fake-accounts/facebook.
18. Zhang, K.; Cao, Q.; Sun, F.; Wu, Y.; Tao, S.; Shen, H.; Cheng, X. Robust Recommender System: A Survey and Future Directions. *ACM Computing Surveys (CSUR)* **2025**.
19. Nguyen, T.T.; Hung, N.Q.V.; Nguyen, T.T.; Huynh, T.T.; Nguyen, T.T.; Weidlich, M.; Yin, H. Manipulating Recommender Systems: A Survey of Poisoning Attacks and Countermeasures. *ACM Computing Surveys (CSUR)* **2025**, *57*, 3:1–3:39.
20. Song, H.; Kim, M.; Park, D.; Shin, Y.; Lee, J.G. Learning From Noisy Labels With Deep Neural Networks: A Survey. *IEEE TNNLS* **2022**, pp. 8135–8153.
21. Pang, G.; Shen, C.; Cao, L.; van den Hengel, A. Deep Learning for Anomaly Detection: A Review. *ACM Computing Surveys (CSUR)* **2022**, *54*, 38:1–38:38.
22. Tang, J.; Wang, K. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. In Proceedings of the WSDM, 2018, pp. 565–573.
23. Kang, W.C.; McAuley, J. Self-attentive sequential recommendation. In Proceedings of the ICDM, 2018, pp. 197–206.
24. Rendle, S.; Freudenthaler, C.; Gantner, Z.; Schmidt-Thieme, L. BPR: Bayesian personalized ranking from implicit feedback. In Proceedings of the UAI, 2009, pp. 452–461.
25. Sun, Y.; Yang, X.; Sun, Z.; Wang, Y.; Wang, B.; Qu, X. LLM4RSR: Large Language Models as Data Correctors for Robust Sequential Recommendation. In Proceedings of the AAAI, 2025, pp. 12604–12612.
26. Jeon, H.; Yoon, S.; McAuley, J.J. Calibration-Disentangled Learning and Relevance-Prioritized Reranking for Calibrated Sequential Recommendation. In Proceedings of the CIKM, 2024, pp. 973–982.
27. Zhao, X.; Zhu, Z.; Caverlee, J. Rabbit Holes and Taste Distortion: Distribution-Aware Recommendation with Evolving Interests. In Proceedings of the TheWebConf, 2021, pp. 888–899.
28. Kaplan, Y.; Krasne, N.; Shtoff, A.; Somekh, O. Unbiased Filtering of Accidental Clicks in Verizon Media Native Advertising. In Proceedings of the CIKM, 2021, pp. 3878–3887.

 doi:10.20944/preprints202601.2218.v1

29. Tolomei, G.; Lalmas, M.; Farahat, A.; Haines, A. You must have clicked on this ad by mistake! Data-driven identification of accidental clicks on mobile ads with applications to advertiser cost discounting and click-through rate prediction. *International Journal of Data Science and Analytics* **2019**, *7*, 53–66.
30. Yao, F.; Li, C.; Nekipelov, D.; Wang, H.; Xu, H. Learning the Optimal Recommendation from Explorative Users. In Proceedings of the AAAI, 2022, pp. 9457–9465.
31. He, J.; Liu, H. Mining Exploratory Behavior to Improve Mobile App Recommendations. *ACM TOIS* **2017**, *35*, 32:1–32:37.
32. Chen, M.; Wang, Y.; Xu, C.; Le, Y.; Sharma, M.; Richardson, L.; Wu, S.; Chi, E.H. Values of User Exploration in Recommender Systems. In Proceedings of the RecSys, 2021, pp. 85–95.
33. Li, N.; Ban, X.; Ling, C.; Gao, C.; Hu, L.; Jiang, P.; Gai, K.; Li, Y.; Liao, Q. Modeling User Fatigue for Sequential Recommendation. In Proceedings of the SIGIR, 2024, pp. 996–1005.
34. Shah, S.S.; Asghar, Z. Dynamics of social influence on consumption choices: A social network representation. *Heliyon* **2023**, *9*.
35. Yoo, H.; Qiu, R.; Xu, C.; Wang, F.; Tong, H. Generalizable Recommender System During Temporal Popularity Distribution Shifts. In Proceedings of the SIGKDD, 2025, pp. 1833–1843.
36. Wan, Z.; Liu, X.; Wang, B.; Qiu, J.; Li, B.; Guo, T.; Chen, G.; Wang, Y. Spatio-temporal contrastive learning-enhanced GNNs for session-based recommendation. *ACM TOIS* **2023**, *42*, 1–26.
37. Zhang, J.; Zhao, Z.; Li, C.; Yu, Y. Lightweight Yet Fine-Grained: A Graph Capsule Convolutional Network with Subspace Alignment for Shared-Account Sequential Recommendation. In Proceedings of the AAAI, 2025, pp. 13242–13250.
38. Zhang, S.; Yin, H.; Chen, T.; Huang, Z.; Nguyen, Q.V.H.; Cui, L. PipAttack: Poisoning Federated Recommender Systems for Manipulating Item Promotion. In Proceedings of the WSDM, 2022, pp. 1415–1423.
39. Anelli, V.W.; Deldjoo, Y.; Noia, T.D.; Merra, F.A. Adversarial Recommender Systems: Attack, Defense, and Advances. In *Recommender Systems Handbook*; 2022; pp. 335–379.
40. Wikipedia. GDPR fines and notices, 2024. https://en.wikipedia.org/wiki/GDPR_fines_and_notices.
41. Cosley, D.; Lam, S.K.; Albert, I.; Konstan, J.A.; Riedl, J. Is seeing believing? How recommender system interfaces affect users' opinions. In Proceedings of the CHI, 2003, pp. 585–592.
42. Lin, Y.; Wang, C.; Chen, Z.; Ren, Z.; Xin, X.; Yan, Q.; de Rijke, M.; Cheng, X.; Ren, P. A Self-Correcting Sequential Recommender. In Proceedings of the CIKM, 2023, pp. 1283–1293.
43. Fan, Z.; Liu, Z.; Wang, Y.; Wang, A.; Nazari, Z.; Zheng, L.; Peng, H.; Yu, P.S. Sequential Recommendation via Stochastic Self-Attention. In Proceedings of the TheWebConf, 2022, pp. 2036–2047.
44. Zhang, S.; Yao, D.; Zhao, Z.; Chua, T.; Wu, F. CauseRec: Counterfactual User Sequence Synthesis for Sequential Recommendation. In Proceedings of the SIGIR, 2021, pp. 367–377.
45. Hsiao, S.; Tsai, Y.; Li, C. Unsupervised Post-Time Fake Social Message Detection with Recommendation-aware Representation Learning. In Proceedings of the TheWebConf, 2022, pp. 232–235.
46. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and Harnessing Adversarial Examples. In Proceedings of the ICLR, 2015.
47. Goldberger, J.; Ben-Reuven, E. Training deep neural-networks using a noise adaptation layer. In Proceedings of the ICLR, 2017.
48. Jindal, I.; Nokleby, M.; Chen, X. Learning deep networks from noisy labels with dropout regularization. In Proceedings of the ICDM, 2016, pp. 967–972.
49. Han, B.; Yao, J.; Niu, G.; Zhou, M.; Tsang, I.; Zhang, Y.; Sugiyama, M. Masking: A new perspective of noisy supervision. In Proceedings of the NeurIPS, 2018.
50. Yao, J.; Wang, J.; Tsang, I.W.; Zhang, Y.; Sun, J.; Zhang, C.; Zhang, R. Deep learning from noisy image labels with quality embedding. *IEEE TIP* **2018**, pp. 1909–1922.
51. YouTube for Press, 2024. https://blog.youtube/press/.
52. Amazon Statistics: Key Numbers and Fun Facts, 2024. https://amzscout.net/blog/amazon-statistics/.
53. Han, B.; Yao, Q.; Yu, X.; Niu, G.; Xu, M.; Hu, W.; Tsang, I.; Sugiyama, M. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In Proceedings of the NeurIPS, 2018, pp. 8536–8546.
54. Yu, X.; Han, B.; Yao, J.; Niu, G.; Tsang, I.; Sugiyama, M. How does disagreement help generalization against label corruption? In Proceedings of the ICML, 2019, pp. 7164–7173.
55. Lin, Y.; Wang, C.; Chen, Z.; Ren, Z.; Xin, X.; Yan, Q.; de Rijke, M.; Cheng, X.; Ren, P. A Self-Correcting Sequential Recommender. In Proceedings of the TheWebConf, 2023, pp. 1283–1293.
56. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. In Proceedings of the ICLR, 2015.

 doi:10.20944/preprints202601.2218.v1

57. Salton, G.; Lesk, M.E. Computer Evaluation of Indexing and Text Processing. *Journal of the ACM* **1968**, *15*, 8–36.
58. Li, J.; Ren, P.; Chen, Z.; Ren, Z.; Lian, T.; Ma, J. Neural Attentive Session-based Recommendation. In Proceedings of the CIKM, 2017, pp. 747–755.
59. Liu, Q.; Zeng, Y.; Mokhosi, R.; Zhang, H. STAMP: short-term attention/memory priority model for session-based recommendation. In Proceedings of the SIGKDD, 2018, pp. 1831–1839.
60. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the NeurIPS, 2017, pp. 5998–6008.
61. Sun, F.; Liu, J.; Wu, J.; Pei, C.; Lin, X.; Ou, W.; Jiang, P. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In Proceedings of the CIKM, 2019, pp. 1441–1450.
62. Fan, Z.; Liu, Z.; Wang, S.; Zheng, L.; Yu, P.S. Modeling Sequences as Distributions with Uncertainty for Sequential Recommendation. In Proceedings of the CIKM, 2021, pp. 3019–3023.
63. Zhang, Y.; Wang, X.; Chen, H.; Zhu, W. Adaptive Disentangled Transformer for Sequential Recommendation. In Proceedings of the SIGKDD, 2023, pp. 3434–3445.
64. Zhou, P.; Ye, Q.; Xie, Y.; Gao, J.; Wang, S.; Kim, J.B.; You, C.; Kim, S. Attention Calibration for Transformer-based Sequential Recommendation. In Proceedings of the CIKM, 2023, pp. 3595–3605.
65. Yuan, J.; Song, Z.; Sun, M.; Wang, X.; Zhao, W.X. Dual Sparse Attention Network For Session-based Recommendation. In Proceedings of the AAAI, 2021, pp. 4635–4643.
66. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the ICML, 2010, pp. 807–814.
67. He, Z.; Zhao, H.; Lin, Z.; Wang, Z.; Kale, A.; McAuley, J.J. Locker: Locally Constrained Self-Attentive Sequential Recommendation. In Proceedings of the CIKM, 2021, pp. 3088–3092.
68. Chen, H.; Lin, Y.; Pan, M.; Wang, L.; Yeh, C.M.; Li, X.; Zheng, Y.; Wang, F.; Yang, H. Denoising Self-Attentive Sequential Recommendation. In Proceedings of the RecSys, 2022, pp. 92–101.
69. Yao, Z.; Chen, X.; Wang, S.; Dai, Q.; Li, Y.; Zhu, T.; Long, M. Recommender Transformers with Behavior Pathways. In Proceedings of the TheWebConf, 2024, pp. 3643–3654.
70. Luo, Z.; Sheng, Z.; Zhang, T. Dual perspective denoising model for session-based recommendation. *ESWA* **2024**, *249*, 123845.
71. Wang, X.; Chen, H.; Pan, Z.; Zhou, Y.; Guan, C.; Sun, L.; Zhu, W. Automated Disentangled Sequential Recommendation with Large Language Models. *ACM TOIS* **2025**, *43*, 29:1–29:29.
72. Pei, W.; Yang, J.; Sun, Z.; Zhang, J.; Bozzon, A.; Tax, D.M.J. Interacting Attention-gated Recurrent Networks for Recommendation. In Proceedings of the CIKM, 2017, pp. 1459–1468.
73. Liu, C.; Li, X.; Cai, G.; Dong, Z.; Zhu, H.; Shang, L. Noninvasive Self-attention for Side Information Fusion in Sequential Recommendation. In Proceedings of the AAAI, 2021, pp. 4249–4256.
74. Liu, H.; Zhu, Y.; Wu, Z. Knowledge Graph-Based Behavior Denoising and Preference Learning for Sequential Recommendation. *IEEE TKDE* **2024**, *36*, 2490–2503.
75. Weston, J.; Chopra, S.; Bordes, A. Memory Networks. In Proceedings of the ICLR, 2015.
76. Chen, X.; Xu, H.; Zhang, Y.; Tang, J.; Cao, Y.; Qin, Z.; Zha, H. Sequential Recommendation with User Memory Networks. In Proceedings of the WSDM, 2018, pp. 108–116.
77. Ma, C.; Ma, L.; Zhang, Y.; Sun, J.; Liu, X.; Coates, M. Memory Augmented Graph Neural Networks for Sequential Recommendation. In Proceedings of the AAAI, 2020, pp. 5045–5052.
78. Tan, Q.; Zhang, J.; Liu, N.; Huang, X.; Yang, H.; Zhou, J.; Hu, X. Dynamic Memory based Attention Network for Sequential Recommendation. In Proceedings of the AAAI, 2021, pp. 4384–4392.
79. Hu, Y.; Liu, Y.; Miao, C.; Miao, Y. Memory Bank Augmented Long-tail Sequential Recommendation. In Proceedings of the CIKM, 2022, pp. 791–801.
80. Huang, J.; Zhao, W.X.; Dou, H.; Wen, J.R.; Chang, E.Y. Improving sequential recommendation with knowledge-enhanced memory networks. In Proceedings of the SIGIR, 2018, pp. 505–514.
81. Qu, S.; Yuan, F.; Guo, G.; Zhang, L.; Wei, W. CmnRec: Sequential Recommendations With Chunk-Accelerated Memory Network. *IEEE TKDE* **2023**, *35*, 3540–3550.
82. Lu, H.; Chai, Z.; Zheng, Y.; Chen, Z.; Xie, D.; Xu, P.; Zhou, X.; Wu, D. Large Memory Network for Recommendation. In Proceedings of the TheWebConf, 2025, pp. 1162–1166.
83. Gers, F.A.; Schmidhuber, J.; Cummins, F.A. Learning to Forget: Continual Prediction with LSTM. *Neural Computation* **2000**, *12*, 2451–2471.
84. Ma, C.; Kang, P.; Liu, X. Hierarchical gating networks for sequential recommendation. In Proceedings of the SIGKDD, 2019, pp. 825–833.

 doi:10.20944/preprints202601.2218.v1

85. Tang, J.; Belletti, F.; Jain, S.; Chen, M.; Beutel, A.; Xu, C.; H Chi, E. Towards neural mixture recommender for long range dependent user sequences. In Proceedings of the TheWebConf, 2019, pp. 1782–1793.
86. Zhao, K.; Zhang, Y.; Yin, H.; Wang, J.; Zheng, K.; Zhou, X.; Xing, C. Discovering Subsequence Patterns for Next POI Recommendation. In Proceedings of the IJCAI, 2020, pp. 3216–3222.
87. Wang, M.; Zhang, S.; Guo, R.; Wang, W.; Wei, X.; Liu, Z.; Yin, H.; Chang, Y.; Zhao, X. STAR-Rec: Making Peace with Length Variance and Pattern Diversity in Sequential Recommendation. In Proceedings of the SIGIR, 2025, pp. 1530–1540.
88. Ma, M.; Ren, P.; Lin, Y.; Chen, Z.; Ma, J.; de Rijke, M. π-Net: A Parallel Information-sharing Network for Shared-account Cross-domain Sequential Recommendations. In Proceedings of the SIGIR, 2019, pp. 685–694.
89. He, Y.; Zhang, Y.; Liu, W.; Caverlee, J. Consistency-Aware Recommendation for User-Generated Item List Continuation. In Proceedings of the WSDM, 2020, pp. 250–258.
90. Chen, C.; Li, D.; Yan, J.; Yang, X. Modeling Dynamic User Preference via Dictionary Learning for Sequential Recommendation. *IEEE TKDE* **2022**, *34*, 5446–5458.
91. Mi, F.; Faltings, B. Memory Augmented Neural Model for Incremental Session-based Recommendation. In Proceedings of the IJCAI, 2020, pp. 2169–2176.
92. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In Proceedings of the ICLR, 2017.
93. Guo, X.; Shi, C.; Liu, C. Intention Modeling from Ordered and Unordered Facets for Sequential Recommendation. In Proceedings of the TheWebConf, 2020, pp. 1127–1137.
94. Liu, X.; Li, Z.; Gao, Y.; Yang, J.; Cao, T.; Wang, Z.; Yin, B.; Song, Y. Enhancing User Intent Capture in Session-Based Recommendation with Attribute Patterns. In Proceedings of the NeurIPS, 2023.
95. Fu, H.; Qin, Z.; Yang, S.; Zhang, H.; Lu, B.; Li, S.; Huang, T.; Lui, J.C.S. Time Matters: Enhancing Sequential Recommendations with Time-Guided Graph Neural ODEs. In Proceedings of the SIGKDD, 2025, pp. 637–648.
96. Zhang, Z.; Wang, X.; Chen, H.; Li, H.; Zhu, W. Disentangled Dynamic Graph Attention Network for Out-of-Distribution Sequential Recommendation. *ACM TOIS* **2025**, *43*, 19:1–19:42.
97. Zhang, S.; Jiang, T.; Kuang, K.; Feng, F.; Yu, J.; Ma, J.; Zhao, Z.; Zhu, J.; Yang, H.; Chua, T.; et al. SLED: Structure Learning based Denoising for Recommendation. *ACM TOIS* **2023**, *42*, 43:1–43:31.
98. Ye, Y.; Xia, L.; Huang, C. Graph Masked Autoencoder for Sequential Recommendation. In Proceedings of the SIGIR, 2023, pp. 321–330.
99. Wang, Z.; Zhu, Y.; Wang, C.; Zhao, X.; Li, B.; Yu, J.; Tang, F. Graph Diffusion-Based Representation Learning for Sequential Recommendation. *IEEE TKDE* **2024**, *36*, 8395–8407.
100. Zeng, X.; Li, S.; Zhang, Z.; Jin, L.; Guo, Z.; Wei, K. RAIN: Reconstructed-aware in-context enhancement with graph denoising for session-based recommendation. *Elsevier Neural Networks* **2025**, *184*, 107056.
101. Sang, C.; Gong, M.; Liao, S.; Zhou, W. MA-GCL4SR: Improving Graph Contrastive Learning-Based Sequential Recommendation with Model Augmentation. *ACM TKDD* **2025**, *19*, 1–21.
102. Gabor, D. Theory of communication. *Journal of the Institution of Electrical Engineers* **1946**, *93*, 429–441.
103. Zhou, K.; Yu, H.; Zhao, W.X.; Wen, J.R. Filter-enhanced MLP is All You Need for Sequential Recommendation. In Proceedings of the TheWebConf, 2022, pp. 2388–2399.
104. Du, X.; Yuan, H.; Zhao, P.; Fang, J.; Liu, G.; Liu, Y.; Sheng, V.S.; Zhou, X. Contrastive Enhanced Slide Filter Mixer for Sequential Recommendation. In Proceedings of the ICDE, 2023, pp. 2673–2685.
105. Shin, Y.; Choi, J.; Wi, H.; Park, N. An Attentive Inductive Bias for Sequential Recommendation beyond the Self-Attention. In Proceedings of the AAAI, 2024, pp. 8984–8992.
106. Xiao, S.; Zhang, J.; Tang, C.; Huang, Z. Frequency-Domain Disentanglement-Fusion and Dual Contrastive Learning for Sequential Recommendation. In Proceedings of the CIKM, 2025, p. 3498–3508.
107. Long, H.; Huang, B.; Lu, J. Learnable Filter with Decoupling Fusion Method for Sequential Recommendation. In Proceedings of the SMC, 2024, pp. 2486–2492.
108. Han, Y.; Wang, H.; Wang, K.; Wu, L.; Li, Z.; Guo, W.; Liu, Y.; Lian, D.; Chen, E. Efficient Noise-Decoupling for Multi-Behavior Sequential Recommendation. In Proceedings of the TheWebConf, 2024, pp. 3297–3306.
109. Wang, H.; Han, Y.; Wang, K.; Cheng, K.; Wang, Z.; Guo, W.; Liu, Y.; Lian, D.; Chen, E. Denoising Pre-Training and Customized Prompt Learning for Efficient Multi-Behavior Sequential Recommendation. *CoRR* **2024**, *abs/2408.11372*.
110. Yu, D.; Lv, C.; Du, X.; Jiang, L.; Yin, Q.; Tong, W.; Zheng, X.; Deng, S. Cost-Effective On-Device Sequential Recommendation with Spiking Neural Networks. In Proceedings of the IJCAI, 2025, pp. 3579–3587.

 doi:10.20944/preprints202601.2218.v1

111. Xia, J.; Li, D.; Gu, H.; Lu, T.; Zhang, P.; Shang, L.; Gu, N. Oracle-guided Dynamic User Preference Modeling for Sequential Recommendation. In Proceedings of the WSDM, 2025, pp. 363–372.
112. Kim, H.; Choi, M.; Lee, S.; Baek, I.; Lee, J. DIFF: Dual Side-Information Filtering and Fusion for Sequential Recommendation. In Proceedings of the SIGIR, 2025, pp. 1624–1633.
113. Huang, Y.; Lu, J.; Li, K.; Zhang, G. Learning a Wavelet Neural Filter with Mamba for Sequential Recommendation. In Proceedings of the 2025 IEEE Symposium on Computational Intelligence in Image, Signal Processing and Synthetic Media (CISM), 2025, pp. 1–7.
114. Heo, B.; Kim, J. WaveRec: Is Wavelet Transform a Better Alternative to Fourier Transform for Sequential Recommendation? In Proceedings of the Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval (ICTIR), 2025, p. 497–502.
115. Xu, H.; Yuan, H.; Liu, G.; Fang, J.; Zhao, L.; Zhao, P. Wavelet Enhanced Adaptive Frequency Filter for Sequential Recommendation. In Proceedings of the AAAI, 2026.
116. Du, X.; Yuan, H.; Zhao, P.; Qu, J.; Zhuang, F.; Liu, G.; Liu, Y.; Sheng, V.S. Frequency Enhanced Hybrid Attention Network for Sequential Recommendation. In Proceedings of the SIGIR, 2023, pp. 78–88.
117. Baek, I.; Yoon, M.; Park, S.; Lee, J. MUFFIN: Mixture of User-Adaptive Frequency Filtering for Sequential Recommendation. In Proceedings of the CIKM, 2025.
118. Su, Y.; Cai, X.; Li, T. FICLRec: Frequency enhanced intent contrastive learning for sequential recommendation. *Elsevier IPM* **2025**, *62*, 104231.
119. Xu, X.; Wang, H.; Guo, W.; Zhang, L.; Yang, W.; Yu, R.; Liu, Y.; Lian, D.; Chen, E. Multi-granularity Interest Retrieval and Refinement Network for Long-Term User Behavior Modeling in CTR Prediction. In Proceedings of the SIGKDD, 2025, pp. 2745–2755.
120. Li, Z.; Sun, A.; Li, C. Diffurec: A diffusion model for sequential recommendation. *ACM TOIS* **2023**, *42*, 1–28.
121. Hou, Y.; Park, J.; Shin, W. Collaborative Filtering Based on Diffusion Models: Unveiling the Potential of High-Order Connectivity. In Proceedings of the SIGIR, 2024, pp. 1360–1369.
122. Wang, Y.; Liu, Z.; Yang, L.; Yu, P.S. Conditional Denoising Diffusion for Sequential Recommendation. In Proceedings of the PAKDD, 2024, Vol. 14649, pp. 156–169.
123. Ma, H.; Xie, R.; Meng, L.; Yang, Y.; Sun, X.; Kang, Z. SeeDRec: Sememe-based Diffusion for Sequential Recommendation. In Proceedings of the IJCAI, 2024, pp. 2270–2278.
124. Cui, X.; Lu, W.; Tong, Y.; Li, Y.; Zhao, Z. Multi-Modal Multi-Behavior Sequential Recommendation with Conditional Diffusion-Based Feature Denoising. In Proceedings of the SIGIR, 2025, pp. 1593–1602.
125. Mao, W.; Yang, Z.; Wu, J.; Liu, H.; Yuan, Y.; Wang, X.; He, X. Addressing Missing Data Issue for Diffusion-based Recommendation. In Proceedings of the SIGIR, 2025, pp. 2152–2161.
126. Ma, H.; Xie, R.; Meng, L.; Chen, X.; Zhang, X.; Lin, L.; Kang, Z. Plug-In Diffusion Model for Sequential Recommendation. In Proceedings of the AAAI, 2024, pp. 8886–8894.
127. Liu, F.; Zou, L.; Zhao, X.; Tang, M.; Dong, L.; Luo, D.; Luo, X.; Li, C. Flow Matching Based Sequential Recommender Model. In Proceedings of the IJCAI, 2025, pp. 3108–3116.
128. Chen, J.; Xu, Y.; Jiang, Y. Unlocking the Power of Diffusion Models in Sequential Recommendation: A Simple and Effective Approach. In Proceedings of the SIGKDD, 2025, p. 155–166.
129. Cai, Z.; Wang, S.; Chu, V.W.; Naseem, U.; Wang, Y.; Chen, F. Unleashing the Potential of Diffusion Models Towards Diversified Sequential Recommendations. In Proceedings of the SIGIR, 2025, pp. 1476–1486.
130. Mao, W.; Liu, S.; Liu, H.; Liu, H.; Li, X.; Hu, L. Distinguished Quantized Guidance for Diffusion-based Sequence Recommendation. In Proceedings of the TheWebConf, 2025, pp. 425–435.
131. Sun, Y.; Wang, B.; Sun, Z.; Yang, X. Does Every Data Instance Matter? Enhancing Sequential Recommendation by Eliminating Unreliable Data. In Proceedings of the IJCAI, 2021, pp. 1579–1585.
132. Sun, Y.; Yang, X.; Sun, Z.; Wang, B. BERD+: A Generic Sequential Recommendation Framework by Eliminating Unreliable Data with Item- and Attribute-level Signals. *ACM TOIS* **2024**, *42*, 41:1–41:33.
133. Zhang, K.; Cao, Q.; Wu, Y.; Sun, F.; Shen, H.; Cheng, X. Personalized Denoising Implicit Feedback for Robust Recommender System. In Proceedings of the TheWebConf, 2025, pp. 4470–4481.
134. Xin, H.; Xiong, Q.; Liu, Z.; Mei, S.; Yan, Y.; Yu, S.; Wang, S.; Gu, Y.; Yu, G.; Xiong, C. ConsRec: Denoising Sequential Recommendation through User-Consistent Preference Modeling. *CoRR* **2025**, *abs/2505.22130*.
135. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *MIT Press JMLR* **2020**, *21*, 1–67.
136. Zhang, K.; Cao, Q.; Wu, Y.; Sun, F.; Shen, H.; Cheng, X. LoRec: Combating Poisons with Large Language Model for Robust Sequential Recommendation. In Proceedings of the SIGIR, 2024, pp. 1733–1742.

 doi:10.20944/preprints202601.2218.v1

137. Yin, M.; Wang, H.; Guo, W.; Liu, Y.; Zhang, S.; Zhao, S.; Lian, D.; Chen, E. Dataset Regeneration for Sequential Recommendation. In Proceedings of the SIGKDD, 2024, pp. 3954–3965.
138. Wang, B.; Liu, F.; Zhang, C.; Chen, J.; Wu, Y.; Zhou, S.; Lou, X.; Wang, J.; Feng, Y.; Chen, C.; et al. LLM4DSR: Leveraging Large Language Model for Denoising Sequential Recommendation. *ACM TOIS* **2025**, *44*.
139. Sun, Y.; Yang, X.; Sun, Z.; Wang, Y.; Wang, B.; Qu, X. LLM4RSR: Large Language Models as Data Correctors for Robust Sequential Recommendation. In Proceedings of the AAAI, 2025, pp. 12604–12612.
140. Wu, T.; Wang, Y.; Wang, M.; Zhang, C.; Zhao, X. Empowering Denoising Sequential Recommendation with Large Language Model Embeddings. In Proceedings of the CIKM, 2025.
141. Hyun, D.; Park, C.; Cho, J.; Yu, H. Beyond Learning from Next Item: Sequential Recommendation via Personalized Interest Sustainability. In Proceedings of the CIKM, 2022, pp. 812–821.
142. Liu, X.; Lu, X.; Cao, Y.; Li, Y. Augmenting Short Sequence for Robust Session-based Recommendation. In Proceedings of the 2024 5th International Conference on Computers and Artificial Intelligence Technology (CAIT), 2024, pp. 260–265.
143. Bacciu, A.; Siciliano, F.; Tonellotto, N.; Silvestri, F. Integrating Item Relevance in Training Loss for Sequential Recommender Systems. In Proceedings of the RecSys, 2023, pp. 1114–1119.
144. Wu, Z.; Wang, X.; Chen, H.; Li, K.; Han, Y.; Sun, L.; Zhu, W. Diff4Rec: Sequential Recommendation with Curriculum-scheduled Diffusion Augmentation. In Proceedings of the ACM MM, 2023, pp. 9329–9335.
145. Liu, Q.; Yan, F.; Zhao, X.; Du, Z.; Guo, H.; Tang, R.; Tian, F. Diffusion Augmentation for Sequential Recommendation. In Proceedings of the CIKM, 2023, pp. 1576–1586.
146. Zhang, C.; Han, Q.; Chen, R.; Zhao, X.; Tang, P.; Song, H. SSDRec: Self-Augmented Sequence Denoising for Sequential Recommendation. In Proceedings of the ICDE, 2024, pp. 803–815.
147. Sun, X.; Sun, F.; Wang, Y.; Song, S.; Tang, W.; Wang, S. Adaptive in-context expert network with hierarchical data augmentation for sequential recommendation. *Elsevier KBS* **2025**, *326*, 114061.
148. Dang, Y.; Liu, Y.; Yang, E.; Huang, M.; Guo, G.; Zhao, J.; Wang, X. Data Augmentation as Free Lunch: Exploring the Test-Time Augmentation for Sequential Recommendation. In Proceedings of the SIGIR, 2025, pp. 1466–1475.
149. Yue, Z.; Zeng, H.; Kou, Z.; Shang, L.; Wang, D. Defending Substitution-Based Profile Pollution Attacks on Sequential Recommenders. In Proceedings of the RecSys, 2022, pp. 59–70.
150. Zhou, K.; Wang, H.; Wen, J.; Zhao, W.X. Enhancing Multi-View Smoothness for Sequential Recommendation Models. *ACM TOIS* **2023**, *41*, 107:1–107:27.
151. Zhang, K.; Cao, Q.; Wu, Y.; Sun, F.; Shen, H.; Cheng, X. Understanding and Improving Adversarial Collaborative Filtering for Robust Recommendation. In Proceedings of the NeurIPS, 2024.
152. Zhang, K.; Cao, Q.; Wu, Y.; Sun, F.; Shen, H.; Cheng, X. Improving the Shortest Plank: Vulnerability-Aware Adversarial Training for Robust Recommender System. In Proceedings of the RecSys, 2024, pp. 680–689.
153. Qian, F.; Chen, W.; Chen, H.; Cui, Y.; Zhao, S.; Zhang, Y. Understanding the Robustness of Deep Recommendation under Adversarial Attacks. *ACM TKDD* **2025**, *19*, 129:1–129:46.
154. Chen, H.; Zhou, K.; Lai, K.; Hu, X.; Wang, F.; Yang, H. Adversarial Graph Perturbations for Recommendations at Scale. In Proceedings of the SIGIR, 2022, pp. 1854–1858.
155. Qin, Q.; Luo, Y.; Chu, Z. DARTS: A Dual-View Attack Framework for Targeted Manipulation in Federated Sequential Recommendation. *CoRR* **2025**, *abs/2507.01383*.
156. Zhang, J.; Hao, B.; Chen, B.; Li, C.; Chen, H.; Sun, J. Hierarchical Reinforcement Learning for Course Recommendation in MOOCs. In Proceedings of the AAAI, 2019, pp. 435–442.
157. Du, Q.; Yu, L.; Li, H.; Leng, Y.; Ou, N. Denoising-Oriented Deep Hierarchical Reinforcement Learning for Next-Basket Recommendation. In Proceedings of the ICASSP, 2022, pp. 4093–4097.
158. Li, Y.; Xiong, H.; Kong, L.; Zhang, R.; Xu, F.; Chen, G.; Li, M. MHRR: MOOCs Recommender Service With Meta Hierarchical Reinforced Ranking. *IEEE TSC* **2023**, *16*, 4467–4480.
159. Antaris, S.; Rafailidis, D. Sequence Adaptation via Reinforcement Learning in Recommender Systems. In Proceedings of the RecSys, 2021, pp. 714–718.
160. Zhao, P.; Luo, C.; Zhou, C.; Qiao, B.; He, J.; Zhang, L.; Lin, Q. RLNF: Reinforcement Learning based Noise Filtering for Click-Through Rate Prediction. In Proceedings of the SIGIR, 2021, pp. 2268–2272.
161. Wang, P.; Fan, Y.; Xia, L.; Zhao, W.X.; Niu, S.; Huang, J.X. KERL: A Knowledge-Guided Reinforcement Learning Model for Sequential Recommendation. In Proceedings of the SIGIR, 2020, pp. 209–218.
162. Li, K.; Wang, P.; Li, C. Multi-agent rl-based information selection model for sequential recommendation. In Proceedings of the SIGIR, 2022, pp. 1622–1631.

 doi:10.20944/preprints202601.2218.v1

163. Namkoong, H.; Duchi, J.C. Stochastic Gradient Methods for Distributionally Robust Optimization with f-divergences. In Proceedings of the NeurIPS, 2016, pp. 2208–2216.
164. Wen, H.; Yi, X.; Yao, T.; Tang, J.; Hong, L.; Chi, E.H. Distributionally-robust Recommendations for Improving Worst-case User Experience. In Proceedings of the TheWebConf, 2022, pp. 3606–3610.
165. Liao, Y.; Yang, Y.; Hou, M.; Wu, L.; Xu, H.; Liu, H. Mitigating Distribution Shifts in Sequential Recommendation: An Invariance Perspective. In Proceedings of the SIGIR, 2025, pp. 1603–1613.
166. Yang, Z.; He, X.; Zhang, J.; Wu, J.; Xin, X.; Chen, J.; Wang, X. A Generic Learning Framework for Sequential Recommendation with Distribution Shifts. In Proceedings of the SIGIR, 2023, pp. 331–340.
167. Zhou, R.; Wu, X.; Qiu, Z.; Zheng, Y.; Chen, X. Distributionally robust sequential recommendation. In Proceedings of the SIGIR, 2023, pp. 279–288.
168. Hu, K.; Li, L.; Xie, Q.; Liu, J.; Tao, X.; Xu, G. Decoupled Progressive Distillation for Sequential Prediction with Interaction Dynamics. *ACM TOIS* **2024**, *42*, 72:1–72:35.
169. Neupane, K.P.; Zheng, E.; Yu, Q. Evidential Stochastic Differential Equations for Time-Aware Sequential Recommendation. In Proceedings of the NeurIPS, 2024.
170. Lin, X.; Pan, W.; Ming, Z. Towards Interest Drift-driven User Representation Learning in Sequential Recommendation. In Proceedings of the SIGIR, 2025, pp. 1541–1551.
171. Yang, Y.; Huang, C.; Xia, L.; Huang, C.; Luo, D.; Lin, K. Debiased Contrastive Learning for Sequential Recommendation. In Proceedings of the TheWebConf, 2023, pp. 1063–1073.
172. Du, H.; Yuan, H.; Zhao, P.; Wang, D.; Sheng, V.S.; Liu, Y.; Liu, G.; Zhao, L. Feature-Aware Contrastive Learning With Bidirectional Transformers for Sequential Recommendation. *IEEE TKDE* **2024**, *36*, 8192–8205.
173. Zhao, C.; Yang, E.; Liang, Y.; Zhao, J.; Guo, G.; Wang, X. Symmetric Graph Contrastive Learning against Noisy Views for Recommendation. *ACM TOIS* **2025**, *43*, 80:1–80:28.
174. Sang, L.; Huang, M.; Wang, Y.; Zhang, Y.; Wu, X. Bottlenecked Heterogeneous Graph Contrastive Learning for Robust Recommendation. *ACM TOIS* **2025**, *43*, 1–36.
175. Tong, X.; Wang, P.; Li, C.; Xia, L.; Niu, S. Pattern-enhanced Contrastive Policy Learning Network for Sequential Recommendation. In Proceedings of the IJCAI, 2021, pp. 1593–1599.
176. Qin, Y.; Wang, P.; Li, C. The World is Binary: Contrastive Learning for Denoising Next Basket Recommendation. In Proceedings of the SIGIR, 2021, pp. 859–868.
177. Chen, Y.; Liu, Z.; Li, J.; McAuley, J.J.; Xiong, C. Intent Contrastive Learning for Sequential Recommendation. In Proceedings of the TheWebConf, 2022, pp. 2172–2182.
178. He, X.; Wei, T.; He, J. Robust Basket Recommendation via Noise-tolerated Graph Contrastive Learning. In Proceedings of the CIKM, 2023, pp. 709–719.
179. Xie, X.; Sun, F.; Liu, Z.; Wu, S.; Gao, J.; Zhang, J.; Ding, B.; Cui, B. Contrastive learning for sequential recommendation. In Proceedings of the ICDE, 2022, pp. 1259–1273.
180. Qin, X.; Yuan, H.; Zhao, P.; Liu, G.; Zhuang, F.; Sheng, V.S. Intent Contrastive Learning with Cross Subsequences for Sequential Recommendation. In Proceedings of the WSDM, 2024, pp. 548–556.
181. Wang, W.; Ma, J.; Zhang, Y.; Zhang, K.; Jiang, J.; Yang, Y.; Zhou, Y.; Zhang, Z. Intent Oriented Contrastive Learning for Sequential Recommendation. In Proceedings of the AAAI, 2025, pp. 12748–12756.
182. Zhang, K.; Cao, Q.; Sun, F.; Liu, X. AsarRec: Adaptive Sequential Augmentation for Robust Self-supervised Sequential Recommendation, 2025, [arXiv:cs.IR/2512.14047].
183. Qiu, R.; Huang, Z.; Yin, H.; Wang, Z. Contrastive Learning for Representation Degeneration Problem in Sequential Recommendation. In Proceedings of the WSDM, 2022, pp. 813–823.
184. Wang, Y.; Zhang, H.; Liu, Z.; Yang, L.; Yu, P.S. ContrastVAE: Contrastive Variational AutoEncoder for Sequential Recommendation. In Proceedings of the CIKM, 2022, pp. 2056–2066.
185. Xia, L.; Huang, C.; Zhang, C. Self-Supervised Hypergraph Transformer for Recommender Systems. In Proceedings of the SIGKDD, 2022, pp. 2100–2109.
186. Yu, Y.; Liu, Q.; Zhang, K.; Zhang, Y.; Song, C.; Hou, M.; Yuan, Y.; Ye, Z.; Zhang, Z.; Yu, S.L. AdaptSSR: Pre-training User Model with Augmentation-Adaptive Self-Supervised Ranking. In Proceedings of the NeurIPS, 2023.
187. Qin, X.; Yuan, H.; Zhao, P.; Fang, J.; Zhuang, F.; Liu, G.; Liu, Y.; Sheng, V.S. Meta-optimized Contrastive Learning for Sequential Recommendation. In Proceedings of the SIGIR, 2023, pp. 89–98.
188. Fan, Z.; Liu, Z.; Peng, H.; Yu, P.S. Mutual Wasserstein Discrepancy Minimization for Sequential Recommendation. In Proceedings of the TheWebConf, 2023, pp. 1375–1385.
189. Hao, Y.; Zhao, P.; Xian, X.; Liu, G.; Zhao, L.; Liu, Y.; Sheng, V.S.; Zhou, X. Learnable Model Augmentation Contrastive Learning for Sequential Recommendation. *IEEE TKDE* **2024**, *36*, 3963–3976.

190. Chen, X.; Wang, Z.; Xu, H.; Zhang, J.; Zhang, Y.; Zhao, W.X.; Wen, J. Data Augmented Sequential Recommendation Based on Counterfactual Thinking. *IEEE TKDE* **2023**, *35*, 9181–9194.
191. Wang, Z.; Zhang, J.; Xu, H.; Chen, X.; Zhang, Y.; Zhao, W.X.; Wen, J. Counterfactual Data-Augmented Sequential Recommendation. In Proceedings of the SIGIR, 2021, pp. 347–356.
192. Chen, J.; Guan, H.; Li, H.; Zhang, F.; Huang, L.; Pang, G.; Jin, X. PACIFIC: Enhancing Sequential Recommendation via Preference-aware Causal Intervention and Counterfactual Data Augmentation. In Proceedings of the CIKM, 2024, pp. 249–258.
193. Tang, S.; Lin, S.; Ma, J.; Zhang, X. CoDeR: Counterfactual Demand Reasoning for Sequential Recommendation. In Proceedings of the AAAI, 2025, pp. 12649–12657.
194. Lin, S.; Tang, S.; Zhang, X.; Ma, J.; Wang, Z. CoDeR+: Interest-aware Counterfactual Reasoning for Sequential Recommendation. *ACM TOIS* **2025**. Just Accepted.
195. Liu, X.; Yuan, J.; Zhou, Y.; Li, J.; Huang, F.; Ai, W. CSRec: Rethinking Sequential Recommendation from A Causal Perspective. In Proceedings of the SIGIR, 2025, pp. 1562–1571.
196. Zhou, H.; Xu, J.; Zhu, Q.; Liu, C. Disentangled Graph Debiasing for Next POI Recommendation. In Proceedings of the SIGIR, 2025, pp. 1779–1788.
197. Fan, L.; Pu, J.; Zhang, R.; Wu, X. Neighborhood-based Hard Negative Mining for Sequential Recommendation. In Proceedings of the SIGIR, 2023, pp. 2042–2046.
198. Kim, K.; Hyun, D.; Yun, S.; Park, C. MELT: Mutual Enhancement of Long-Tailed User and Item for Sequential Recommendation. In Proceedings of the SIGIR, 2023, pp. 68–77.
199. Zhang, C.; Du, Y.; Zhao, X.; Han, Q.; Chen, R.; Li, L. Hierarchical Item Inconsistency Signal Learning for Sequence Denoising in Sequential Recommendation. In Proceedings of the CIKM, 2022, pp. 2508–2518.
200. Bian, S.; Zhao, W.X.; Zhou, K.; Cai, J.; He, Y.; Yin, C.; Wen, J. Contrastive Curriculum Learning for Sequential User Behavior Modeling via Data Augmentation. In Proceedings of the CIKM, 2021, pp. 3737–3746.
201. Chen, H.; Chen, Y.; Wang, X.; Xie, R.; Wang, R.; Xia, F.; Zhu, W. Curriculum Disentangled Recommendation with Noisy Multi-feedback. In Proceedings of the NeurIPS, 2021, pp. 26924–26936.
202. Wang, Y.; Ge, X.; Chen, X.; Xie, R.; Yan, S.; Zhang, X.; Chen, Z.; Ma, J.; Xin, X. Exploration and Exploitation of Hard Negative Samples for Cross-Domain Sequential Recommendation. In Proceedings of the WSDM, 2025, pp. 669–677.
203. Steck, H. Calibrated recommendations. In Proceedings of the RecSys, 2018, pp. 154–162.
204. Seymen, S.; Abdollahpouri, H.; Malthouse, E.C. A Constrained Optimization Approach for Calibrated Recommendations. In Proceedings of the RecSys, 2021, pp. 607–612.
205. Chambolle, A. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision* **2004**, *20*, 89–97.
206. Zhao, X.; Zhu, Z.; Caverlee, J. Rabbit Holes and Taste Distortion: Distribution-Aware Recommendation with Evolving Interests. In Proceedings of the TheWebConf, 2021, pp. 888–899.
207. Abdollahpouri, H.; Nazari, Z.; Gain, A.; Gibson, C.; Dimakopoulou, M.; Anderton, J.; Carterette, B.A.; Lalmas, M.; Jebara, T. Calibrated Recommendations as a Minimum-Cost Flow Problem. In Proceedings of the WSDM, 2023, pp. 571–579.
208. Jeon, H.; Yoon, S.; McAuley, J.J. Calibration-Disentangled Learning and Relevance-Prioritized Reranking for Calibrated Sequential Recommendation. In Proceedings of the CIKM, 2024, pp. 973–982.
209. Chen, J.; Wu, W.; Shi, L.; Ji, Y.; Hu, W.; Chen, X.; Zheng, W.; He, L. DACSR: Decoupled-Aggregated End-to-End Calibrated Sequential Recommendation. *Springer Applied Intelligence* **2022**.
210. Chen, J.; Wu, W.; Shi, L.; Zheng, W.; He, L. Long-tail session-based recommendation from calibration. *Springer Applied Intelligence* **2023**, pp. 4685–4702.
211. Lesota, O.; Bajko, A.; Walder, M.; Wenzel, M.; Tommasel, A.; Schedl, M. Fine-tuning for Inference-efficient Calibrated Recommendations. In Proceedings of the RecSys, 2025, pp. 1187–1192.
212. Li, C.; Liu, Z.; Wu, M.; Xu, Y.; Zhao, H.; Huang, P.; Kang, G.; Chen, Q.; Li, W.; Lee, D.L. Multi-Interest Network with Dynamic Routing for Recommendation at Tmall. In Proceedings of the CIKM, 2019, pp. 2615–2623.
213. Cen, Y.; Zhang, J.; Zou, X.; Zhou, C.; Yang, H.; Tang, J. Controllable Multi-Interest Framework for Recommendation. In Proceedings of the SIGKDD, 2020, pp. 2942–2951.
214. LaLonde, R.; Bagci, U. Capsules for Object Segmentation. In Proceedings of the arXiv preprint arXiv:1804.04241, 2018.

 doi:10.20944/preprints202601.2218.v1

215. Wang, S.; Hu, L.; Wang, Y.; Sheng, Q.Z.; Orgun, M.A.; Cao, L. Modeling Multi-Purpose Sessions for Next-Item Recommendations via Mixture-Channel Purpose Routing Networks. In Proceedings of the IJCAI, 2019, pp. 3771–3777.
216. Li, S.; Yang, D.; Zhang, B. MRIF: Multi-resolution Interest Fusion for Recommendation. In Proceedings of the SIGIR, 2020, pp. 1765–1768.
217. Tian, Y.; Chang, J.; Niu, Y.; Song, Y.; Li, C. When Multi-Level Meets Multi-Interest: A Multi-Grained Neural Model for Sequential Recommendation. In Proceedings of the SIGIR, 2022, pp. 1632–1641.
218. Lee, J.; Yun, J.; Kang, U. BaM: AnEnhanced Training Scheme for Balanced and Comprehensive Multi-interest Learning. In Proceedings of the SIGKDD, 2024.
219. Chen, W.; Ren, P.; Cai, F.; Sun, F.; de Rijke, M. Improving End-to-End Sequential Recommendations with Intent-aware Diversification. In Proceedings of the CIKM, 2020, pp. 175–184.
220. Chen, W.; Ren, P.; Cai, F.; Sun, F.; De Rijke, M. Multi-interest diversification for end-to-end sequential recommendation. *ACM TOIS* **2021**, *40*, 1–30.
221. Li, B.; Jin, B.; Song, J.; Yu, Y.; Zheng, Y.; Zhou, W. Improving Micro-video Recommendation via Contrastive Multiple Interests. In Proceedings of the SIGIR, 2022, pp. 2377–2381.
222. Zhang, S.; Yang, L.; Yao, D.; Lu, Y.; Feng, F.; Zhao, Z.; Chua, T.; Wu, F. Re4: Learning to Re-contrast, Re-attend, Re-construct for Multi-interest Recommendation. In Proceedings of the TheWebConf, 2022, pp. 2216–2226.
223. Wang, C.; Wang, Z.; Liu, Y.; Ge, Y.; Ma, W.; Zhang, M.; Liu, Y.; Feng, J.; Deng, C.; Ma, S. Target Interest Distillation for Multi-Interest Recommendation. In Proceedings of the CIKM, 2022, pp. 2007–2016.
224. Xie, Y.; Gao, J.; Zhou, P.; Ye, Q.; Hua, Y.; Kim, J.B.; Wu, F.; Kim, S. Rethinking Multi-Interest Learning for Candidate Matching in Recommender Systems. In Proceedings of the RecSys, 2023, pp. 283–293.
225. Du, Y.; Wang, Z.; Sun, Z.; Ma, Y.; Liu, H.; Zhang, J. Disentangled Multi-interest Representation Learning for Sequential Recommendation. In Proceedings of the SIGKDD, 2024, pp. 677–688.
226. Wang, J.; Zeng, Z.; Wang, Y.; Wang, Y.; Lu, X.; Li, T.; Yuan, J.; Zhang, R.; Zheng, H.; Xia, S. MISSRec: Pre-training and Transferring Multi-modal Interest-aware Sequence Representation for Recommendation. In Proceedings of the ACM MM, 2023, pp. 6548–6557.
227. Liu, Y.; Zhang, X.; Zou, M.; Feng, Z. Co-occurrence Embedding Enhancement for Long-tail Problem in Multi-Interest Recommendation. In Proceedings of the RecSys, 2023, pp. 820–825.
228. Yan, J.; Jiang, L.; Cui, J.; Zhao, Z.; Bin, X.; Zhang, F.; Liu, Z. Trinity: Syncretizing Multi-/Long-Tail/Long-Term Interests All in One. In Proceedings of the SIGKDD, 2024, pp. 6095–6104.
229. Zheng, Y.; Wang, G.; Liu, Y.; Lin, L. Diversity Matters: User-Centric Multi-Interest Learning for Conversational Movie Recommendation. In Proceedings of the ACM MM, 2024, pp. 9515–9524.
230. Liu, Y.; Zhang, X.; Zou, M.; Feng, Z. Attribute Simulation for Item Embedding Enhancement in Multi-interest Recommendation. In Proceedings of the WSDM, 2024, pp. 482–491.
231. Hu, S.; Wu, W.; Tang, Z.; Huan, Z.; Wang, L.; Zhang, X.; Zhou, J.; Zou, L.; Li, C. HORAE: Temporal Multi-Interest Pre-training for Sequential Recommendation. *ACM TOIS* **2025**, *43*, 88:1–88:29.
232. Karypis, G. Evaluation of item-based top-n recommendation algorithms. In Proceedings of the CIKM, 2001, pp. 247–254.
233. Liu, T.Y. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval* **2009**, *3*, 225–331.
234. Jarvelin, K.; Kekalainen, J. IR evaluation methods for retrieving highly relevant documents. In Proceedings of the SIGIR, 2000, pp. 41–48.
235. Zhao, Y.; Wang, Y.; Liu, Y.; Cheng, X.; Aggarwal, C.C.; Derr, T. Fairness and diversity in recommender systems: a survey. *ACM TIST* **2025**, *16*, 1–28.
236. Winter, L. Dataset Pruning in RecSys and ML: Best Practice or Mal-Practice? *arXiv preprint arXiv:2510.14704* **2025**.
237. Ma, H.; Xie, R.; Meng, L.; Feng, F.; Du, X.; Sun, X.; Kang, Z.; Meng, X. Negative sampling in recommendation: A survey and future directions. *arXiv preprint arXiv:2409.07237* **2024**.
238. Krichene, W.; Rendle, S. On Sampled Metrics for Item Recommendation. In Proceedings of the SIGKDD, 2020, pp. 1748–1757.
239. Bousquet, O.; Elisseeff, A. Algorithmic Stability and Generalization Performance. In Proceedings of the NeurIPS, 2000, pp. 196–202.
240. Zhang, S.; Yao, D.; Zhao, Z.; Chua, T.S.; Wu, F. Causerec: Counterfactual user sequence synthesis for sequential recommendation. In Proceedings of the SIGIR, 2021, pp. 367–377.

 doi:10.20944/preprints202601.2218.v1

241. Chen, Z.; Deng, Y.; Wu, Y.; Gu, Q.; Li, Y. Towards Understanding the Mixture-of-Experts Layer in Deep Learning. In Proceedings of the NeurIPS, 2022, pp. 23049–23062.