

Article

A Mixture-of-Experts Model for Improved Generalization in Session-Aware Recommendation

Sungshin Kwak ¹, Jaedong Lee ^{2,*} and Sohyun Park ^{2,*} 

¹ Department of AI Convergence, Dankook University, Yongin 16890, Republic of Korea; sungshin98@dankook.ac.kr

² Department of Computer Science, Dankook University, Yongin 16890, Republic of Korea

* Correspondence: letsdoit@dankook.ac.kr (J.L.); sohyunpark@dankook.ac.kr (S.P.); Tel.: +82-31-8005-3254 (J.L. & S.P.)

Abstract

Recently, recommendation systems have actively integrated Transformers to capture real-time context. However, these systems often suffer from generalization imbalance, where predictions are biased toward popular (head) items due to the sparsity and volatility inherent in session-based data. To address this challenge, this paper proposes MoE-SLMRec, a Mixture-of-Experts (MoE)-based recommendation model that selects expert networks based on session-level contextual information. The proposed model extracts a session latent representation, h , through a session-aware controller and forms balanced predictive characteristics across the entire data distribution via dynamic routing. Experimental results demonstrate that MoE-SLMRec significantly outperforms the baseline SLMRec, improving accuracy by 1.51 percentage points (from 18.76% to 20.27%). Furthermore, the model achieved state-of-the-art performance in Recall@20 (0.8358) and MRR@20 (0.3455), validating simultaneous improvements in both retrieval capability and ranking quality. Notably, the model effectively stabilized the performance for head items while coordinating the generalization trade-off between head and tail segments. By ensuring a favorable capacity–cost trade-off while maintaining robust performance, this study presents a promising alternative under session-based recommendation settings, facilitating scalable deployment in real-time recommendation services.

Keywords: head–tail distribution; load balancing; mixture-of-experts; recommender systems; sequential modeling; session-based recommendation



Academic Editors: Yanping Zhang, Jianjun Yang and Wenlin Han

Received: 9 January 2026

Revised: 10 February 2026

Accepted: 13 February 2026

Published: 14 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Recently, recommendation systems have become essential decision-making tools across e-commerce, media streaming, advertising, and content platforms [1]. Consequently, the scale and complexity of user behavior logs have grown exponentially. In real-world service environments, where individual user interests often fluctuate within short timeframes, there is active research into models capable of effectively capturing sequential behavioral contexts. In this context, Transformer-based models [2–4] are widely utilized for their ability to effectively model complex intra-session interactions. However, session-based recommendation environments are characterized by limited observable interactions and high volatility in input distributions [5]. As a result, existing models tend to exhibit biased fitting toward specific frequency domains or repetitive behavioral patterns rather than generalizing in a balanced manner across the entire data distribution. Such bias leads to

conservative predictions skewed toward either popular (head) items [6,7] or simplistic long-tail patterns, ultimately resulting in a generalization imbalance across the global data distribution [8].

This imbalance stems from the inherent constraints of single-model architectures in effectively capturing diverse user behavior patterns with varying frequency and complexity. Accordingly, there is a technical demand for architectures that can decouple these disparate patterns—either orthogonally or independently—to enhance expressiveness and regulate generalization characteristics. The Mixture-of-Experts (MoE) framework has emerged as a prominent solution to address this need, selectively activating only a subset of expert networks based on input [9,10]. This allows for an effective expansion of model capacity while maintaining controlled computational costs [11,12]. Given these efficiencies, MoE has served as a core architecture for scaling expressiveness in large-scale models, and its potential application in session-based recommendation is currently being actively explored [13].

However, a naïve application of the MoE framework is insufficient to fundamentally resolve the generalization imbalance in session-based environments. Under sparse feedback and short-session constraints, expert concentration may occur, where activation becomes excessively skewed toward specific experts [10]. This leads to a scenario where learning is reinforced primarily for high-frequency behavioral patterns, while relatively rare patterns are insufficiently captured [14,15]. This phenomenon results in a structural trade-off between the predictive performance of Head and Tail items, acting as a significant constraint in securing the stable recommendation quality required for real-world services. Ultimately, conventional monolithic modeling approaches are limited in their ability to simultaneously accommodate these polarized data characteristics, necessitating a flexible response system that avoids bias toward specific domains [16,17].

Driven by this insight, this study focuses on balancing the generalization characteristics between Head and Tail segments as the fundamental key to overall performance enhancement. Accordingly, we propose MoE-SLMRec, a MoE-based recommendation model that performs adaptive expert selection based on session-level context [18,19]. MoE-SLMRec is derived by extending the original SLMRec [20] architecture, in which the core session encoding backbone is preserved while the feed-forward transformation is reformulated as a Mixture-of-Experts module. By retaining the sequential modeling capability of SLMRec and augmenting it with expert-level specialization, the proposed model enables dynamic decomposition of complex intra-session behaviors without introducing additional sequential paths.

Rather than locally optimizing for specific data segments, the proposed model aims to secure robust predictive performance across the entire data distribution through dynamic routing that reflects the temporal dynamics of session units [21]. MoE-SLMRec primarily consists of a Session-aware Controller, Expert Networks, and a Routing Mechanism [22]. The session-aware controller generates a latent representation that represents individual user sessions, serving as the core input signal for expert selection [23]. Each expert is trained to specialize in distinct user behavioral patterns, and the routing mechanism facilitates contextual expert selection to mitigate excessive bias and enhance the model's overall generalization performance.

The effectiveness of the proposed approach is validated through experiments. MoE-SLMRec improved overall recommendation accuracy from 18.76% to 20.27% compared to the baseline SLMRec, marking a performance gain of 1.51 percentage points. In recommendation environments characterized by high volatility, an improvement exceeding 1% is considered a substantial gain from a practical perspective [24]. Notably, the model stabilized and enhanced the predictive performance for Head items, suggesting that com-

plex intra-session patterns can be effectively decoupled at the expert level [14]. While a slight performance shift was observed in the Tail segment, this is interpreted as a re-configuration of the generalization boundary, an inherent trade-off in the specialization process [15]. Our approach to Balanced Generalization does not aim for the independent maximization of all segments but involves strategically coordinating characteristics based on overall system utility. This forms an optimal equilibrium between the conservative stability of single-path models and the expanded expressiveness of MoE. Consequently, this study presents a realistic and attractive alternative for constructing large-scale, real-time recommendation systems where accurate prediction of high-value items directly influences business outcomes [11].

2. Related Work

2.1. Session-Based Recommendation Models

The effectiveness of session-based recommendation is the task of predicting the next item to be selected by utilizing only a sequence of interactions that occurred over a short period, particularly in scenarios where explicit long-term user preference information is unavailable [25]. In these environments, stable generalization is notoriously difficult due to the limited contextual information within individual sessions and the high volatility of user behavior. Early research primarily focused on modeling intra-session sequential dependencies using Recurrent Neural Networks (RNNs), such as GRU4Rec [25]. This approach served as a catalyst for addressing session-based recommendation through the lens of neural network-based sequence modeling.

Following the emergence of the Transformer architecture [26], self-attention-based models like SASRec were proposed [27,28]. These models demonstrated consistent performance improvements over RNN-based models by effectively capturing long-range dependencies and complex interactions between items. In particular, the self-attention mechanism allows for the parallel modeling of relationships between items within a session, offering the advantage of flexibly learning diverse behavioral patterns. These developments have further highlighted the importance of sequence representation learning in session-based recommendation.

More recently, research has actively expanded this trend by applying large-scale pre-training techniques [29] and language model structures to recommendation systems [30]. For instance, SLMRec [4] treats user sessions as linguistic sequences and significantly improves the generalization performance of session representations by applying a Transformer-based encoder. This approach excels over previous models in its ability to effectively share common behavioral patterns across sessions.

However, existing models still face limitations regarding generalization on imbalanced data distributions, particularly for long-tail items [6]. In real-world recommendation logs, interactions tend to be heavily concentrated on popular (head) items. Consequently, models preferentially adapt to high-frequency patterns, failing to acquire sufficient representational capacity for Tail items with lower observation frequencies. This results in a structurally fixed generalization gap between head and tail items, even when overall performance appears stable [31].

To overcome these limitations, this paper proposes replacing the Feed-Forward Network (FFN) of the existing SLMRec with a Mixture-of-Experts (MoE) architecture [10]. Furthermore, we design a Session-aware Controller that performs expert selection by explicitly reflecting the characteristics of the input session. This enables the decoupled modeling of user behavior patterns with varying frequency and complexity at the session level. The ultimate goal is to move beyond static learning of generalization characteristics and secure structural flexibility that can dynamically adjust based on the session context.

2.2. Mixture-of-Experts (MoE)

Mixture-of-Experts (MoE) is an architecture that expands model capacity by partitioning the internal structure into multiple Expert Networks and selectively activating only a subset of experts based on input characteristics [9,10], enabling flexible trade-offs between capacity and computational cost. Unlike single-path models that use the same computational path for all inputs, MoE can simultaneously achieve high expressiveness and computational efficiency by dynamically configuring different paths for different inputs [12]. Initially proposed as a means to maximize the expressiveness of large-scale neural networks, MoE was later formally introduced into large-scale Transformer structures, primarily within the field of Natural Language Processing (NLP).

A prominent example is Google's Switch Transformer, which replaced the Transformer's FFN layers with multiple experts and utilized a Top-k routing strategy to activate only a few experts per input token. This demonstrated that the total number of parameters could be significantly expanded while maintaining computational costs similar to conventional Transformers. Subsequently, Google DeepMind's GLaM implemented a massive language model with hundreds of billions of parameters using an MoE structure, empirically proving that MoE can practically support the scalability of large language models [32]. These studies indicate that MoE has evolved beyond a simple structural variation into a core paradigm for large-scale model design.

However, MoE architectures inherently face the Routing Collapse problem [33], where selection becomes excessively concentrated on specific experts depending on the distribution of input data and the routing mechanism. This phenomenon can lead to a situation where only a few experts are repeatedly activated while the remaining experts are not sufficiently trained. To mitigate this, various regularization techniques have been proposed to control the distribution of expert utilization. Representative methods such as Load-balancing loss, Entropy regularization, and Temperature scaling primarily aim to prevent excessive concentration by inducing a uniform selection probability across experts.

Nevertheless, such control methods based on a uniform distribution assumption may have limitations in environments where input distributions are highly imbalanced, such as in session-based recommendation [34]. In actual user behavior logs, popular items and specific behavioral patterns are observed with overwhelming frequency, and session lengths are mostly short and heterogeneous. In such environments, allowing experts to naturally specialize in different behavioral patterns and item distributions may actually contribute more to performance improvement [19]. Therefore, forcibly equalizing the utilization frequency of all experts may inhibit role differentiation among experts and consequently restrict the model's expressiveness [35].

Based on this problem awareness, this study adopts Lightweight Load-balancing Regularization, which minimizes routing collapse instead of imposing strong load-balancing constraints that target equal utilization of all experts. The proposed regularization term aims to secure training stability while maintaining natural role differentiation by applying penalties only when excessive concentration toward specific experts occurs. Through this, we aim to implement an MoE routing structure that is more suitable for the session-based recommendation environment.

While MoE architectures have demonstrated strong scalability and adaptability in large-scale sequence modeling tasks, their application to session-based recommendation remains relatively underexplored. In particular, it remains unclear how expert routing mechanisms can be adapted to handle recommendation-specific challenges such as heterogeneous session structures and popularity imbalance. To better position our work within existing research directions, the following section compares representative session-based

recommendation and MoE approaches and highlights the methodological gaps addressed in this study.

2.3. Methodological Comparison and Research Gap

While session-based recommendation models and Mixture-of-Experts architectures have both demonstrated strong performance in their respective domains, their methodological focuses differ. Session-based recommendation approaches mainly concentrate on improving sequence representation learning within sessions, whereas MoE-based methods primarily aim to increase model capacity and computational scalability through dynamic expert routing. However, limited attention has been given to combining these two directions to explicitly address generalization challenges arising from popularity imbalance in session-based recommendation.

In session-based recommendation, Transformer-based models such as SASRec and SLMRec effectively capture sequential interaction patterns and improve session representation quality. Nevertheless, these models typically rely on a single shared feed-forward transformation for all sessions, making it difficult to adapt model capacity to sessions with heterogeneous behavioral complexity or item popularity distributions.

On the other hand, MoE-based approaches successfully introduce expert specialization by dynamically routing inputs to different expert networks. However, existing MoE architectures are primarily designed for scalability in large language models or general sequence processing tasks. Their routing and load-balancing mechanisms typically aim to enforce uniform expert utilization rather than addressing recommendation-specific data imbalance, such as the head–tail popularity skew observed in real-world recommendation logs.

To clarify methodological differences and highlight the remaining gaps, Table 1 summarizes representative approaches and their limitations. The comparison shows that prior session-based models lack adaptive expert routing mechanisms, while existing MoE methods do not explicitly consider session characteristics or recommendation-oriented popularity imbalance.

Table 1. Gap comparison table.

Method	Session Modeling	MoE Routing	Load Balancing	Session-Aware Routing	Head–Tail Handling	Limitation/Gap
GRU4Rec	✓ (RNN)	✗	✗	✓	✗	No expert specialization; popularity bias persists
SASRec	✓ (Transformer)	✗	✗	partial	✗	No adaptive routing or expert modeling
SLMRec	✓ (Transformer)	✗	✗	✓	partial	No expert routing or load control
Switch Transformer/ MoE models	✗	✓	✓	✗	✗	Designed for scalability, not recommendation imbalance
MoE-SLMRec (Ours)	✓	✓	✓ (RecLB)	✓	✓	—

The proposed MoE-SLMRec bridges these gaps by integrating session-aware expert routing with recommendation-aware load-balancing regularization, enabling expert specialization while controlling routing collapse under imbalanced interaction distributions. This design allows the model to adaptively allocate representational capacity according

to session characteristics and item frequency patterns, improving head–tail generalization balance without sacrificing overall recommendation quality.

3. Materials and Methods

3.1. Problem Formulation

Session-based recommendation environments inherently suffer from structural challenges in achieving balanced generalization across the entire data distribution, due to the short and sparsely observed nature of user interactions [36]. In particular, Recent Transformer-based recommendation models and Mixture-of-Experts (MoE) architectures exhibit strong advantages in terms of representational capacity expansion. However, they also tend to adapt preferentially to high-frequency items or repetitive user behavior patterns. While this tendency can improve prediction performance for popular (Head) items, it often induces a trade-off in generalization performance between Head and Tail items from a global data distribution perspective.

Recognizing that completely eliminating this trade-off is impractical in real-world recommendation settings, this study adopts a more pragmatic viewpoint. Rather than independently maximizing performance in specific data segments, we argue that strategically regulating the generalization characteristics between Head and Tail items based on overall recommendation utility is a more effective approach. Accordingly, our objective is to recalibrate the generalization boundary of MoE-based recommendation models by incorporating session-level contextual information into the expert selection process, enabling more balanced and stable performance across the distribution.

3.2. Proposed Model: MoE-SLMRec

The MoE-SLMRec model proposed in this study is designed to improve overall recommendation accuracy by balancing Head and Tail items while maintaining the capacity expansion capabilities of the Mixture-of-Experts (MoE) architecture within a session-based recommendation environment. The proposed model distinguishes itself from conventional single-path recommendation models or naïve MoE applications. Specifically, by performing expert routing based on session-level contextual information, it effectively mitigates performance bias toward specific data domains or high-frequency patterns. The overall architecture is illustrated in Figure 1.

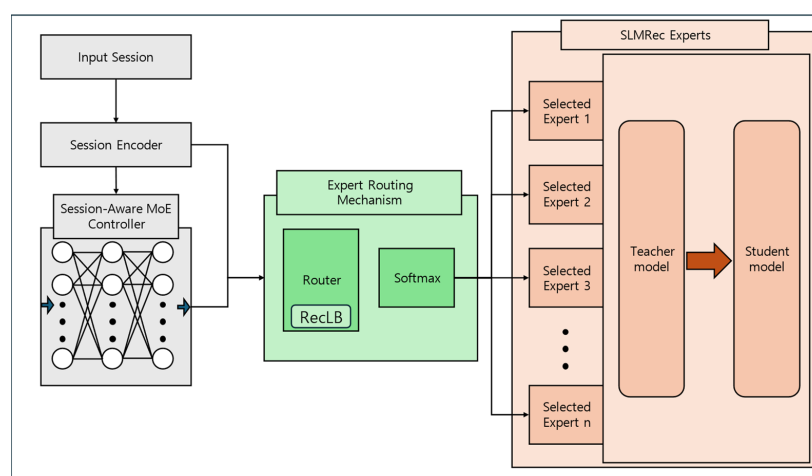


Figure 1. MoE-SLMRec structure diagram. Dynamic expert selection and knowledge transfer are performed through session-aware controllers and RecLB-based routing mechanisms.

As shown in Figure 1, the MoE-SLMRec architecture consists of three primary components: the Session-aware MoE Controller, the Set of Expert Networks, and the Expert Routing Mechanism.

The Session-aware MoE Controller takes the interaction sequence within a user session as input and generates a latent representation that summarizes the context of the current session. This representation serves as the core control signal for selecting the experts to be activated. By handling decision-making at the session level, the controller enables consistent, non-fragmented expert selection, even in environments characterized by short interactions and sparse feedback.

This automated routing strategy is also consistent with recent studies highlighting the importance of automated decision-making mechanisms in complex data-driven industrial systems. In practical deployment environments, manually designed or static configurations often struggle to adapt to highly dynamic and high-dimensional user behavior patterns. Automated approaches, such as those explored in AutoML-based system [37] design, reduce reliance on manual configuration and improve robustness by enabling models to adapt automatically to changing data characteristics. Similarly, the proposed session-aware controller performs automatic expert routing decisions based on session context, allowing the model to dynamically adjust expert utilization without requiring manual intervention or fixed routing strategies, thereby facilitating scalable deployment in real-world recommendation services.

The Set of Expert Networks comprises multiple networks trained to adapt to diverse user behavioral patterns and item distributions. While each expert shares the same input structure, the training process induces natural Specialization toward different data characteristics. This ensures that the expansion of expressiveness provided by the MoE structure translates into substantial performance gains by capturing various interaction features, rather than mere parameter scaling.

The Expert Routing Mechanism receives the session representation from the controller to select active experts and assign weights. Instead of token-level routing, our model adopts a session-unit routing strategy and incorporates an internal RecLB (Recommendation Load-Balancing) module to prevent Routing Collapse (over-concentration on specific experts). This mechanism facilitates more stable and generalized predictive characteristics across the entire data distribution.

The final recommendation score is obtained by aggregating the outputs of the selected experts using a routing-weighted combination. Knowledge distillation, when employed, is applied during the backbone training stage to obtain an efficient session encoder, and no additional teacher–student distillation objective is used in MoE-SLMRec training. The ultimate goal is to achieve generalization centered on holistic performance—carefully managing the trade-off between Head and Tail items—rather than localized optimization for specific data segments.

The detailed procedure for MoE-SLMRec is presented in Algorithm 1. As an extension that integrates MoE into the SLMRec framework, MoE-SLMRec maintains the fundamental architecture of the original SLMRec while enhancing its representational flexibility.

The proposed algorithm preserves the overall architecture of the original SLMRec while replacing the Feed-Forward Network (FFN) layers with a Mixture-of-Experts (MoE) structure, enabling adaptive expert selection based on session context.

First, in Line 2, a batch-level set of sessions S is transformed into a session-level embedding h through the session encoder and pooling operation employed in SLMRec. This process is identical to that of the original SLMRec and aggregates item order and interaction information within each session into a compact representation.

Algorithm 1. MoE-SLMRec

Input $S, y, k, \lambda_1, \lambda_2$
Output Θ

1. Def Moe-SLMRec(S, y):
2. $h \leftarrow \text{SLMRec.SessionEncoder.Pool}(S)$
3. $z, T \leftarrow \text{MoEController}(h)$
4. $g \leftarrow \text{Softmax}(z / T)$
5. $\text{Idx}, g' \leftarrow \text{TopK}(g, k); g_{\text{norm}} \leftarrow \text{Renorm}(g')$
6. $h_{\text{moe}} = \sum_{i \in \text{Idx}} g_{\text{norm}}^{(i)} \cdot \text{FFN}_i(h)$
7. $\text{logits} \leftarrow \text{SLMRec.OutputHead}(h_{\text{moe}})$
8. $\mathcal{L} \leftarrow \text{CE}(\text{logits}, y) + \lambda_1 \mathcal{L}_{\text{LB}}(g) + \lambda_2 \mathcal{L}_{\text{Temp}}(T)$

The parameters for this are shown in Table 2.

Table 2. Parameters of Algorithm 1.

Variable Name	Description
S	A batch-level set of user sessions.
y	Ground-truth target items for each session.
k	Number of Top-k experts selected for computation in the MoE layer.
λ_1, λ_2	Hyperparameters that adjust the relative weights of the load-balancing loss and the temperature regularization loss, respectively.
θ	The set of all trainable network weights updated during the learning process.
h	A compressed latent vector representing the overall context of a session via the SessionEncoder.
z	Initial scores generated by the MoEController for expert selection.
T	Temperature parameter that controls the smoothness of the routing distribution; higher values encourage diverse expert selection, while lower values lead to more concentrated routing.
g	Probability distribution for expert selection calculated via the Softmax function.
idx	Indices of the Top-k experts with the highest selection probabilities in g .
g_{norm}	Normalized weights for the Top-k experts, re-scaled to sum to 1.
h_{moe}	The final aggregated output vector, calculated as the weighted sum of the selected experts' outputs.
logits	Raw recommendation scores for all candidate items before probability transformation.
$\mathcal{L}$	The total objective function value to be minimized during model training.

Next, in Line 3, the MoE controller takes the session embedding h as input and predicts a temperature parameter T , which controls the sensitivity of expert selection, along with a logit adjustment vector z applied to the gating outputs. Through this mechanism, the expert selection distribution is dynamically adapted according to session characteristics.

In Line 4, the gating network computes selection scores for each expert based on the session embedding. A softmax operation is then applied to the temperature-scaled and logit-adjusted scores, producing the expert selection probability distribution g . This distribution reflects the relative importance of each expert for the current session.

Subsequently, the Top-k experts with the highest probabilities in g are selected, and their corresponding weights are renormalized. This strategy enables the model to leverage a small subset of experts that are most relevant to the session while maintaining computational efficiency by avoiding activation of all experts.

In Line 5, the selected experts replace the FFN layer of the original SLMRec. The output of each expert is combined through a weighted linear aggregation using the renormalized

weights g_{norm} , forming the MoE-based representation h_{moe} . This process enhances model expressiveness by allowing different combinations of experts to be activated across sessions.

Finally, in Line 7, the aggregated MoE representation h_{moe} is passed through the output head of SLMRec to produce next-item prediction scores, logits. During training, the final loss L is computed by jointly optimizing the standard cross-entropy loss for recommendation, a recommendation-aware load-balancing regularization term L_{LB} to mitigate excessive concentration of expert usage, and a temperature regularization term L_{Temp} to stabilize the learned temperature parameter.

MoE-SLMRec builds upon an SLMRec-style backbone. While backbone models can optionally be pretrained using distillation-based procedures, distillation is not part of the proposed MoE-SLMRec training objective. Instead, this work focuses on architectural modification through MoE-based expert routing, and the training objective includes only the standard recommendation loss and routing regularization.

3.3. Session-Aware MoE Controller

The session-aware MoE controller plays a central role in encoding an input user session into a latent representation space, summarizing the overall behavioral characteristics of the current session. Given an interaction sequence that constitutes a session, the proposed model learns a fixed-length session representation vector through a DNN-based encoder. This controller is designed to capture not only simple item frequency information, but also the sequential context and interaction patterns within the session in an integrated manner.

Specifically, each item interaction is first mapped to a low-dimensional embedding vector and aggregated through a multi-layer perceptron (MLP)-based encoder. Through nonlinear transformations, the encoder abstracts complex interaction relationships among items within the session and produces a d -dimensional latent session representation vector h . The encoder is implemented as a two-layer MLP with hidden dimension d , ReLU activation, and dropout rate p .

In session-based recommendation environments, interaction-level information is often sparse and highly volatile, making expert selection at the token level prone to unnecessary routing fluctuations. In particular, when sessions are short or user behavior is irregular, token-level routing may cause unstable expert activation patterns, hindering consistent expert specialization. In contrast, the latent session representation h , which aggregates the entire sequence of interactions within a session, provides a more stable summary of overall behavioral patterns, contextual signals, and interaction complexity. Routing decisions based on this aggregated session representation therefore reflect session-level intent rather than momentary interaction variations, leading to more stable and consistent expert selection. Furthermore, since sessions differ in length, behavioral diversity, and item frequency distribution, session-level routing naturally enables experts to specialize in different session characteristics. For these reasons, we adopt the session-level latent representation h , rather than interaction-level representations, as the routing input, since it provides a more suitable and stable basis for expert selection in session-based recommendation settings.

Considering the strict inference latency constraints in real-world recommendation services, this study adopts a shallow DNN architecture with limited depth for the controller encoder. This design choice prevents session representation learning from becoming a computational bottleneck in the overall recommendation pipeline, while keeping the computational overhead manageable within the overall recommendation pipeline.

In addition, the controller focuses exclusively on encoding session characteristics, while the actual expert activation decisions are deferred to the subsequent expert routing mechanism. This clear separation of responsibilities allows the proposed model to independently optimize session representation learning and expert selection. As a result, the

model achieves more stable and precise expert routing even in environments characterized by short interaction sequences and sparse feedback.

After obtaining the session representation h , the controller produces two outputs used for expert routing: an expert logit adjustment vector z and a temperature parameter T . The vector z modulates expert preference scores, while the temperature parameter controls the sharpness of the routing distribution when computing gating weights via softmax normalization.

To maintain routing stability, the predicted temperature is constrained within a predefined valid range so that excessively small temperature values do not cause expert over-concentration. Specifically, temperature values are bounded using a scaled sigmoid function to ensure they remain within this predefined range. Furthermore, a temperature regularization loss is introduced during training to discourage unstable routing behavior and maintain balanced expert utilization. This design ensures that expert routing remains stable while preserving sufficient flexibility for session-adaptive specialization.

3.4. Expert Networks

The expert networks constitute a core component of the Mixture-of-Experts (MoE) architecture and consist of a set of expert models $\{E_1, E_2, E_3, \dots, E_n\}$ designed to capture diverse user behavior patterns and item distributions. Rather than introducing novel expert architectures, the proposed model adopts a standard expert design and focuses primarily on the expert selection mechanism. This design choice avoids unnecessary increases in model complexity while effectively maximizing the representational benefits of the MoE framework.

Each expert E_i is implemented as a Feed-Forward Network (FFN) with an identical architecture and receives the session representation h produced by the session-aware MoE controller as input. During training, the experts are encouraged to naturally adapt to different session types and data characteristics. This enables the model to distribute learning across experts, capturing not only frequently observed patterns associated with popular (Head) items but also a diverse range of interaction characteristics present in the data.

Importantly, functional specialization among experts is not explicitly predefined. All experts are trained under the same optimization objective, and their roles emerge in a data-driven manner through the interaction between the session-aware controller and the expert routing mechanism. This design prevents experts from being rigidly assigned to predetermined data regions and allows their functional roles to flexibly evolve in accordance with the observed session distribution during training.

Furthermore, instead of activating all experts simultaneously, the proposed model aggregates the outputs of only the Top- k experts selected by the routing mechanism to produce the final recommendation scores. This strategy ensures computational efficiency while providing expanded representational capacity compared to single-model architectures.

3.5. Expert Routing Mechanism

The expert routing mechanism determines the activation strength of each expert based on the session representation h provided by the session-aware MoE controller. Instead of adopting fine-grained routing at the token or item level, this study employs a session-level routing strategy, which enables more consistent expert selection even in environments characterized by short sessions and sparse feedback. This design mitigates unnecessary fluctuations in expert selection within the same session and promotes stable functional specialization among experts.

Specifically, given the session representation vector h generated by the session-aware controller, the routing function computes a probability distribution g over all experts, where N denotes the total number of experts. The routing (gate) scores are defined as follows:

$$g = \text{softmax}\left(\frac{W_g h + b_g}{T}\right) \quad (1)$$

where W_g and b_g are learnable parameters, and T is a temperature parameter that controls the smoothness of the routing distribution. The resulting weights g indicate the relative contribution of each expert to the final prediction.

For computational efficiency, only the Top-K experts with the highest routing probabilities in g are selected to form an index set I . The corresponding routing weights are then renormalized to obtain g_{norm} , ensuring that the selected weights sum to one.

The final MoE-based representation h_{moe} is computed as a weighted sum of the outputs of the selected experts, as follows:

$$y = \sum_{e=1}^E g_e \cdot y_e, \quad (2)$$

This Top-K sparse activation strategy allows the model to effectively exploit the representational capacity of the MoE architecture without activating all experts simultaneously.

3.6. RecLB (Recommendation Load Balancing)

In session-based MoE recommendation systems, imbalanced expert selection can lead to a routing collapse phenomenon, where traffic becomes excessively concentrated on a small subset of experts. This issue is particularly pronounced during the early stages of training, where certain experts may monopolize high-frequency (Head) sessions. As a result, other experts receive limited learning signals, which distorts expert specialization and degrades overall model robustness. To mitigate this issue, we introduce RecLB (Recommendation Load Balancing Loss), a recommendation-aware regularization term that explicitly accounts for the data distribution characteristics of recommendation systems.

RecLB begins by defining the expert usage distribution. Let B denote the batch size and N the total number of experts. For a session S_b , let g_e^b denote the gating probability that expert e is selected. The average usage (importance) of expert e over a batch, denoted as f_e , is calculated as:

$$f_e = \frac{1}{B} \sum_{b=1}^B g_e^b \quad (3)$$

This quantity represents the proportion of overall session traffic assigned to expert e . To alleviate basic expert collapse, RecLB introduces a usage balancing term that penalizes only extreme concentration in expert usage, defined as:

$$\mathcal{L}_{\text{usage}} = N \sum_{e=1}^N (f_e)^2 \quad (4)$$

This term does not aim to enforce uniform expert utilization, but instead applies regularization only when a particular expert becomes excessively dominant.

However, recommendation data inherently follows a long-tail distribution, making simple usage balancing insufficient to address concentration on popular sessions. To account for this, we define a popularity-based weight w_{pop}^b for each session S_b , derived

from the popularity of its ground-truth item. Using this weight, the popularity-weighted expert usage P_{pop} is computed as:

$$P_{pop}(e) = \frac{\sum_{b=1}^B w_{pop}^b \cdot g_e^b}{\sum_{b=1}^B w_{pop}^b} \quad (5)$$

Based on this quantity, the popularity-aware balancing term is defined as:

$$\mathcal{L}_{pop} = N \sum_{e=1}^N (P_{pop}(e))^2 \quad (6)$$

This term suppresses scenarios in which a small number of experts disproportionately process popular (Head) sessions, encouraging Head traffic to be distributed across multiple experts. As a consequence, Tail sessions and items gain increased opportunities to be processed by diverse experts.

In addition, to promote diversity in expert routing paths, we introduce a traffic diversity term that maximizes the entropy of the expert usage distribution:

$$\mathcal{L}_{div} = - \sum_{e=1}^N f_e \log(f_e) \quad (7)$$

This entropy-based regularization increases routing diversity, thereby improving the likelihood that Tail sessions and Tail items are handled by multiple experts. Since $\mathcal{L}_{div}$ corresponds to the entropy of expert utilization, the final objective is formulated such that minimizing the total loss encourages higher entropy, thereby preventing expert over-concentration.

Finally, the overall RecLB loss is defined as a weighted combination of the three components:

$$\mathcal{L}_{RecLB} = \alpha \mathcal{L}_{usage} + \beta \mathcal{L}_{pop} - \gamma \mathcal{L}_{div} \quad (8)$$

where α , β , and γ are hyperparameters controlling the contribution of each term. The negative sign before the entropy term explicitly ensures that minimizing the overall loss promotes higher routing entropy, preventing routing collapse caused by expert over-concentration. Through this design, RecLB effectively mitigates early-stage Head expert lock-in without forcibly constraining expert usage to a uniform $1/N$ distribution. As a result, the proposed approach preserves the representational capacity of the MoE architecture while maintaining a balanced Head–Tail trade-off aligned with recommendation data characteristics. In all experiments, the coefficients are set to $\alpha = 1.0$, $\beta = 1.0$, and $\gamma = 0.1$. This formulation removes ambiguity in the optimization direction of the entropy term and prevents unintended expert concentration during implementation.

4. Experimental Results

4.1. Dataset

In this study, experiments were conducted using the Yoochoose dataset from the RecSys Challenge 2015 [38]. Yoochoose is a representative benchmark dataset for session-based recommendation, consisting of user click logs collected from an e-commerce platform. The dataset is defined solely by item click sequences within individual sessions, without explicit user identifiers. As a result, sessions are relatively short, and the data exhibits a high degree of sparsity, which makes the recommendation task particularly challenging.

The Yoochoose dataset reflects real-world service environments and exhibits an extremely imbalanced long-tail distribution in item popularity. A small number of popular items account for the majority of clicks, while most items are observed only infrequently.

Consequently, popularity bias and performance degradation on Tail items have been recognized as central challenges in recommendation system research using this dataset.

In our experiments, the raw click logs were first sorted by session identifier and timestamp, and sessions containing fewer than two interactions were removed. The remaining sessions were then transformed into item sequences of maximum length T ($T = 20$ in our experiments), and the task was formulated as a next-item prediction problem. The last item in each session was treated as the prediction target (label), while the preceding sequence was used as the model input. For sessions longer than $T + 1$ interactions, only the most recent interactions were retained to construct training samples. The column structure of the preprocessed dataset is summarized in Table 3. The resulting samples were randomly divided into training, validation, and test sets using an 80/10/10 ratio with a fixed random seed to ensure reproducibility, and item popularity statistics were computed using only the training set.

Table 3. Configure the source column of the Yoochoose dataset.

Column Name	Description
Session_id	Identifier for distinguishing one user session
Timestamp	Time information when the user interacted with the item
item_id	Unique identifier of the item clicked within the session
category	Category information to which the item belongs

To facilitate a detailed long-tail analysis and examine the generalization characteristics of the model with respect to item popularity, we partition the entire item set into Head and Tail items. First, item popularity is defined based on the occurrence frequency of each item in the training data. Specifically, the popularity of item i , denoted as $Pop(i)$, is computed as follows:

$$Pop(i) = \sum_{s \in S} \sum_{x \in s} \mathbb{I}(x = i) \quad (9)$$

where S denotes the set of all sessions in the training set, and $\mathbb{I}(\cdot)$ is an indicator function that returns 1 if the condition holds and 0 otherwise.

After computing item frequencies, all items are sorted in descending order of popularity. The top $r = 20\%$ of items are defined as Head items, while the remaining 80% are categorized as Tail items. This partitioning strategy enables a clear distinction between a small subset of highly popular items that account for the majority of click events and a large number of infrequently observed items. As a result, it allows for a quantitative analysis of popularity bias mitigation and region-wise generalization performance of the proposed model. The threshold $r = 20\%$ follows common practice in long-tail recommendation studies and provides a balanced trade-off between coverage and interpretability.

4.2. Implementation Details

All models were trained under identical preprocessing and training configurations to ensure fair comparison. The session dataset was preprocessed and stored in NPZ format for efficient loading, where sessions were converted into fixed-length sequences and randomly split into training, validation, and test sets using an 80/10/10 ratio with a fixed random seed.

Training was conducted using the Adam optimizer with a learning rate of 1×10^{-3} and a batch size of 256 for up to 100 epochs. Model selection was performed based on validation performance, and the checkpoint achieving the best combined Recall@20 and MRR@20 score was used for final evaluation. All methods were evaluated using Top-20 ranking metrics.

Next-item prediction was trained using cross-entropy loss over all candidate items. When load-balancing regularization was enabled, an auxiliary load-balancing loss term was added with a weighting coefficient of 0.01. Hyperparameters shared across models were tuned using the same validation protocol, while parameters specific to the MoE architecture, such as the number of experts and routing configuration, were tuned only within the proposed model framework.

This unified training protocol ensures that performance differences arise from architectural improvements rather than discrepancies in training configurations.

4.3. Model Performance Evaluation

In this section, we analyze the performance of the proposed MoE-SLMRec by comparing it with the baseline SLMRec model and several state-of-the-art (SOTA) recommendation models. All experiments were conducted under identical data preprocessing procedures and training configurations to ensure fair comparisons. The primary differences among the models lie in whether the Feed-Forward Network (FFN) within the Transformer blocks is replaced by a Mixture-of-Experts (MoE) structure and whether the proposed recommendation-aware load-balancing regularization (RecLB) is applied.

Performance comparisons were conducted not only in terms of overall recommendation quality but also with respect to item frequency, considering both Head and Tail item segments. The evaluation metrics used in this study are summarized in Table 4.

Table 4. Performance metrics.

Metrics	Description
Overall ACC	The proportion of test sessions for which the correct target item is successfully recommended, reflecting the overall predictive accuracy of the recommendation system
Head ACC	Accuracy computed only on popular (Head) items with high occurrence frequency in the entire log, evaluating the model's performance on frequent items.
Tail ACC	Accuracy computed only on infrequent (Tail) items, reflecting the model's generalization ability and recommendation diversity in long-tail scenarios.
Recall@20(R@20)	Measures the proportion of sessions in which the ground-truth item appears within the top-20 recommended items, evaluating the candidate retrieval capability of the model.
MRR@20	Measures ranking quality by averaging the reciprocal rank of the ground-truth item within the top-20 recommendation list, reflecting how quickly the correct item is ranked near the top.

Specifically, recommendation performance was evaluated using Overall Accuracy, Head Accuracy, Tail Accuracy, Recall@20 (R@20), and Mean Reciprocal Rank@20 (MRR@20). Recall@20 measures the probability that the item actually selected by the user appears within the top-20 recommended items, reflecting the coverage and correctness of the recommendation list. MRR@20 computes the average reciprocal rank of the ground-truth item within the top-20 recommendations, capturing how quickly a user can reach the desired item.

By jointly considering Overall Accuracy, Head Accuracy, Tail Accuracy, R@20, and MRR@20, we provide a comprehensive evaluation of model performance across both popularity segments and ranking quality. The quantitative results are reported in Table 5.

Table 5. Performance evaluation.

Model	Overall ACC	Head ACC	Tail ACC	R@20	MRR@20
SLMRec	0.1876	0.1461	0.3137	0.8154	0.3252
MoE-SLMRec	0.2027	0.1888	0.2449	0.8358	0.3455
BERT4Rec	0.1915	0.2051	0.1459	0.8040	0.3221
SASRec	0.1784	0.1655	0.2257	0.7856	0.3009
GRU4Rec	0.1601	0.1554	0.2058	0.7555	0.2754

MoE-SLMRec achieved an Overall Accuracy of 0.2027, representing an approximately 8.0% relative improvement over the baseline SLMRec (0.1876). It also consistently outperformed Transformer-based SASRec (0.1784) and RNN-based GRU4Rec (0.1601), demonstrating superior predictive accuracy across different model families. In particular, MoE-SLMRec achieved the highest scores on the Top-K ranking metrics, with Recall@20 of 0.8358 and MRR@20 of 0.3455. These results indicate that the proposed model not only improves the likelihood of including the correct item in the recommendation list, but also places it at higher ranks, thereby achieving superior ranking quality.

A segment-wise analysis based on item popularity reveals that the proposed model yields a substantial performance gain in the Head item region, improving accuracy from 0.1461 to 0.1888 compared to SLMRec. This improvement suggests that the MoE architecture effectively disentangles and models dominant behavioral patterns within sessions, leading to more accurate predictions for frequently occurring items.

In contrast, a performance degradation is observed in the Tail item region, where accuracy decreases from 0.3137 to 0.2449 relative to SLMRec. This phenomenon can be interpreted as a structural generalization trade-off arising from the representational expansion of the MoE architecture, where optimization toward high-frequency patterns is prioritized. Similar tendencies are also observed in other models such as SASRec, reaffirming that accurate prediction of sparse items remains a fundamental challenge in session-based recommendation.

To improve interpretability of the Head–Tail performance comparison, we additionally report the number of test instances in each segment together with segment-wise ranking metrics and their 95% confidence intervals, as summarized in Table 6.

Table 6. Segment-wise ranking metrics.

Segment	Test Cases	Recall@20 (95% CI)	MRR@20 (95% CI)
Head	5322	0.8200 $\pm$ 0.0103	0.3360 $\pm$ 0.0096
Tail	1289	0.8813 $\pm$ 0.0177	0.4070 $\pm$ 0.0205

To provide a finer-grained analysis of performance across item popularity, we further conduct evaluation based on item-frequency deciles. Following the Pareto principle, which commonly characterizes recommendation logs where a small portion of popular items accounts for a large share of interactions, items are first sorted according to training interaction frequency and then partitioned into ten equally sized groups.

For readability, we summarize the decile-level results by grouping them into Head (Deciles 1–3), Middle (Deciles 4–7), and Tail (Deciles 8–10) segments, as shown in Table 7. The results indicate that MoE-SLMRec achieves clear improvements in high-frequency item groups while maintaining competitive performance in lower-frequency segments. This decile-based analysis provides additional quantitative evidence that the proposed routing

and load-balancing mechanism reallocates model capacity across popularity regions rather than uniformly optimizing a single segment, supporting the balanced generalization claim.

Table 7. Performance across item-frequency groups.

Group	Model	Test Cases	Recall@20	MRR@20
Head (Deciles 1–3)	SLMRec	55,314	0.784	0.306
	MoE-SLMRec	55,314	0.82	0.329
Middle (4–7)	SLMRec	6828	0.914	0.485
	MoE-SLMRec	6828	0.872	0.403
Tail (8–10)	SLMRec	2957	0.926	0.535
	MoE-SLMRec	2957	0.899	0.455

For further comparison with a modern Transformer-based session recommendation baseline, we additionally evaluated BERT4Rec under the same experimental protocol. BERT4Rec achieved an Overall Accuracy of 0.1915, which surpasses both GRU4Rec (0.1601) and SASRec (0.1784) while remaining slightly below MoE-SLMRec (0.2027). In terms of ranking quality, BERT4Rec obtained Recall@20 of 0.8040 and MRR@20 of 0.3220, demonstrating competitive ranking capability among Transformer-based session models, although still lower than MoE-SLMRec, which achieved 0.8358 and 0.3455, respectively.

A segment-wise analysis further reveals differences in generalization behavior across item popularity groups. BERT4Rec achieves a relatively high accuracy of 0.2050 in the Head region, exceeding both SLMRec (0.1461) and MoE-SLMRec (0.1888), indicating that bidirectional contextual modeling effectively captures dominant behavioral patterns associated with popular items. However, performance in the Tail region drops to 0.1450, significantly below SLMRec (0.3137) and MoE-SLMRec (0.2449). This observation suggests that while Transformer-based contextual modeling strengthens predictions for frequent interaction patterns, it still struggles to generalize effectively to sparse items.

In contrast, MoE-SLMRec achieves a more balanced outcome by improving overall recommendation accuracy while preserving substantially stronger Tail performance compared with BERT4Rec. These results indicate that performance gains obtained solely through Transformer-based sequence modeling do not fully resolve the Head–Tail generalization imbalance. Instead, the session-aware expert routing and RecLB regularization employed in MoE-SLMRec provide a practical mechanism for coordinating expert specialization, enabling improved overall recommendation utility while maintaining more robust performance across the item popularity spectrum.

Overall, the experimental results demonstrate that MoE-SLMRec does not aim to maximize performance in the Tail region in isolation but instead recalibrates the generalization boundary to maximize overall recommendation utility. Notably, the pronounced performance improvements in the Head region directly translate into gains in ranking-based metrics such as Recall and MRR, which are closely associated with user satisfaction in real-world service environments. Taken together, this study provides an empirical analysis of the performance imbalance introduced by MoE architectures and presents a practical approach for coordinating such trade-offs to improve overall recommendation quality.

To better understand the internal behavior behind these improvements, we further analyze the stability of the expert routing mechanism. Table 8 summarizes expert utilization statistics for head and tail sessions, allowing us to verify that routing does not collapse into a small subset of experts. The results show that expert usage remains broadly distributed, with maximum expert selection shares remaining below 9% and entropy values close to the uniform baseline, indicating stable utilization across experts. In addition, the learned

temperature values remain within a stable range (mean ≈ 0.51 , minimum fixed at the enforced lower bound of 0.5), confirming that routing does not become overly sharp. These observations provide quantitative evidence that the proposed routing mechanism operates stably even under short and volatile session conditions.

Table 8. Routing stability summary table.

Session Type	Entropy	Top-1 Share	Gini
Head	2.76	0.086	0.368
Tail	2.75	0.089	0.335

4.4. Ablation Study

In this section, we conduct an ablation study to analyze the contribution of each component in the proposed MoE-SLMRec model to the overall recommendation performance. Using the baseline SLMRec as Ablation 1, we systematically evaluate performance variations by selectively enabling or disabling the session-aware controller, the Mixture-of-Experts (MoE) architecture, and the proposed RecLB regularization term. The results of these ablation experiments are summarized in Table 9.

Table 9. Ablation study results of MoE-SLMRec. O indicates that the component is enabled, while X indicates that the component is removed or disabled in the ablation configuration.

Ablation	Controller	MoE	RecLB	Overall Acc	Head Acc	Tail Acc	R@20	MRR@20
Ablation 1 (Δ vs. Ablation 2, SLMRec)	X	X	X	−0.0151	−0.0427	+0.0688	−0.0204	−0.0203
Ablation 2 (MoE-SLMRec, baseline)	O	O	O	0.2027	0.1888	0.2449	0.8358	0.3455
Ablation 3 (Δ vs. Ablation 2)	X	O	O	−0.005	−0.0072	−0.002	−0.0525	−0.0195
Ablation 4 (Δ vs. Ablation 2)	X	O	X	−0.0076	−0.0109	−0.005	−0.0548	−0.0155

One of the most notable observations from the ablation results is the severe performance degradation observed when the session-aware controller is removed (Ablations 3 and 4). Replacing the Feed-Forward Network with a Mixture-of-Experts architecture without explicit control signals (Ablation 4), or introducing load-balancing without session-aware guidance (Ablation 3), leads to unstable and inconsistent performance gains, indicating routing inefficiency rather than effective expert specialization. In session-based recommendation settings characterized by short and volatile sequences, MoE routing without explicit guidance struggles to learn meaningful expert allocation.

In contrast, the proposed model integrating all components (Ablation 2) achieves consistent improvements across evaluation metrics. The session-aware controller stabilizes expert routing, while RecLB distributes expert utilization, enabling the increased representational capacity of MoE to translate into measurable recommendation gains. These results indicate that performance improvements arise from the interaction of components rather than isolated module effects.

Consistent with earlier experiments, Tail Accuracy remains highest in the single-path baseline SLMRec. However, the proposed model increases Head Accuracy by +0.0427 while reducing Tail Accuracy by −0.0688, reallocating capacity toward items contributing more to ranking quality. This trade-off ultimately improves overall ranking performance, with MRR@20 increasing by +0.0203. These results quantitatively support the claim that the proposed approach recalibrates generalization toward improved recommendation utility rather than merely amplifying popularity bias.

To further examine whether the observed Head–Tail performance difference depends on routing configuration, we evaluate the trained MoE-SLMRec model using different numbers of activated experts. As shown in Table 10, increasing k leads to only marginal improvements in Tail performance while overall and Head performance remain largely unchanged. This indicates that activating more experts does not fundamentally remove the Head–Tail imbalance. Instead, the observed performance shift reflects an inherent capacity–specialization trade-off in MoE routing rather than a specific routing configuration.

Table 10. Performance variation with activated experts (k).

Model	k	Overall ACC	Head ACC	Tail ACC	R@20	MRR@20
MoE-SLMRec	1	0.2025	0.1893	0.2426	0.8357	0.3453
MoE-SLMRec	2	0.2027	0.1891	0.2439	0.8358	0.3455
MoE-SLMRec	4	0.2027	0.1890	0.2442	0.8358	0.3455

In addition to accuracy analysis, we also report computational cost measurements comparing SLMRec and MoE-SLMRec under identical inference settings. Table 11 summarizes parameter count, inference latency, throughput, and GPU memory consumption. The results show that inference latency increases monotonically with the number of activated experts k , confirming that computational cost is primarily governed by the number of active experts and reflecting the capacity–cost trade-off introduced by MoE routing.

Table 11. Computational cost comparison.

Model	Activated Experts (k)	Parameters	Avg. Latency (ms/Batch)	p95 Latency (ms/Batch)	Throughput (Samples/s)	Peak GPU Memory (MB)
SLMRec (plain)	–	18,798,168	5.74	7.23	44,620	295.08
MoE-SLMRec	1	26,737,401	190.34	224.81	1344.96	295.08
MoE-SLMRec	2	26,737,401	327.81	369.14	780.93	295.08
MoE-SLMRec	4	26,737,401	597.71	704.82	428.3	295.08

As shown in Table 11, inference latency increases monotonically with the number of activated experts k , confirming that computational cost is primarily governed by the number of active experts. This analysis provides a quantitative view of the capacity–cost trade-off introduced by MoE-based expert routing.

5. Conclusions

In this paper, we focus on the Head–Tail generalization imbalance that commonly arises in session-based recommendation settings and propose MoE-SLMRec, a session-aware Mixture-of-Experts (MoE) recommendation model designed to mitigate this issue. Due to the inherent characteristics of session-based recommendation—namely short interaction sequences and sparse feedback—models tend to over-adapt to high-frequency items and repetitive behavioral patterns. This tendency not only limits generalization across the entire data distribution but also poses challenges to the stability and long-term utility of recommendation systems.

Rather than framing this problem solely as a matter of improving Tail item performance, this study redefines it from a more practical perspective: how to strategically balance generalization characteristics between Head and Tail items based on overall recommendation utility. By adopting this viewpoint, we emphasize that meaningful performance gains

in session-based recommendation settings require coordinated trade-offs across different popularity regions, rather than isolated optimization of specific segments.

The proposed MoE-SLMRec leverages session-level contextual information to guide expert selection, effectively adapting the representational advantages of MoE architectures to session-based recommendation environments. In particular, by adopting a session-level routing strategy instead of token- or item-level routing, the model maintains consistent expert selection even under short and volatile input sequences, thereby structurally mitigating routing collapse. Furthermore, the introduction of RecLB (Recommendation Load Balancing Loss) explicitly accounts for the long-tailed nature of recommendation data, suppressing excessive expert monopolization without forcibly enforcing uniform expert utilization.

Experimental results on the Yoochoose dataset demonstrate that MoE-SLMRec improves Overall Accuracy from 0.1876 to 0.2027, achieving an approximately 8.0% relative performance gain over the baseline SLMRec. Consistent improvements are also observed in ranking-based metrics such as Recall@20 and MRR@20, indicating that the proposed model not only increases the likelihood of retrieving the correct item but also ranks it more accurately within the recommendation list. Although a structural trade-off is observed in the Tail region due to the refined modeling of Head items, this behavior reflects a meaningful reallocation of generalization capacity toward overall recommendation quality as reflected by ranking-based metrics.

The key contribution of this work lies in its empirical analysis of the generalization behavior of MoE architectures in session-based recommendation from an item frequency perspective, and in proposing a practical architectural framework to regulate this behavior. The integration of a session-aware controller with recommendation-specific load-balancing regularization provides an effective design paradigm for future MoE-based recommendation systems under session-based recommendation environments characterized by popularity imbalance. While our evaluation was conducted on the Yoochoose benchmark, future work will extend experiments to additional public session-based datasets to further validate the generality of the observed head–tail trade-off patterns.

For future work, we plan to further enhance Tail item generalization by exploring adaptive MoE structures that dynamically adjust the number of active experts based on session characteristics, as well as hybrid expert designs incorporating external metadata and content features. In addition, we aim to evaluate the long-term impact of the proposed model through online experiments and extended user behavior analysis in practical session-based recommendation scenarios.

Author Contributions: Conceptualization, S.K.; methodology, S.K.; formal analysis, S.K.; software, S.K.; validation, S.K.; investigation, J.L. and S.P.; resources, S.K.; supervision, J.L. and S.P.; project administration, S.K.; writing—original draft preparation, S.K.; writing—review and editing, S.K., J.L. and S.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available in <https://www.kaggle.com/code/raceproptit/yoochoose> (accessed on 9 January 2026).

Acknowledgments: During the preparation of this manuscript, the authors used tools to translate Korean text into English. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Feng, L.; Yuan, H.; Ye, Q.; Qian, Y.; Ge, X. Exploring the impacts of a recommendation system on an e-platform based on consumers' online behavioral data. *Inf. Manag.* **2024**, *61*, 103905. [\[CrossRef\]](#)
- Lu, Y.; Gao, Z.; Cheng, Z.; Sun, J.; Brown, B.; Yu, G.; Wong, A.; Pérez, F.; Volkovs, M. Session-based recommendation with transformers. In *Proceedings of the Recommender Systems Challenge 2022*; Association for Computing Machinery: New York, NY, USA, 2022; pp. 29–33.
- Qiao, S.; Zhou, W.; Wen, J.; Gao, C.; Luo, Q.; Chen, P.; Li, Y. Multi-view Intent Learning and Alignment with Large Language Models for Session-based Recommendation. *ACM Trans. Inf. Syst.* **2025**, *43*, 1–25. [\[CrossRef\]](#)
- Su, J.; Zhong, Q.; Ma, Y.; Liu, W.; Chen, C.; Zheng, X.; Yin, J.; Chua, T.-S. Distilling Transitional Pattern to Large Language Models for Multimodal Session-based Recommendation. *arXiv* **2025**, arXiv:2504.10538. [\[CrossRef\]](#)
- Redjidal, A.; Pinto, L.; Desmarais, M. Optimizing Encoder-Only Transformers for Session-Based Recommendation Systems. *arXiv* **2024**, arXiv:2410.11150.
- Liu, S.; Zheng, Y. Long-tail session-based recommendation. In *Proceedings of the 14th ACM Conference on Recommender Systems*, Online, 22–26 September 2020; pp. 509–514.
- Gupta, P.; Garg, D.; Malhotra, P.; Vig, L.; Shroff, G. NISER: Normalized item and session representations to handle popularity bias. *arXiv* **2019**, arXiv:1909.04276.
- Raja, R.; Vats, A.; Vats, A.; Majumder, A. A Comprehensive Review on Harnessing Large Language Models to Overcome Recommender System Challenges. *arXiv* **2025**, arXiv:2507.21117.
- Jacobs, R.A.; Jordan, M.I.; Nowlan, S.J.; Hinton, G.E. Adaptive mixtures of local experts. *Neural Comput.* **1991**, *3*, 79–87. [\[CrossRef\]](#)
- Shazeer, N.; Mirhoseini, A.; Maziarz, K.; Davis, A.; Le, Q.; Hinton, G.; Dean, J. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv* **2017**, arXiv:1701.06538.
- Lepikhin, D.; Lee, H.; Xu, Y.; Chen, D.; Firat, O.; Huang, Y.; Krikun, M.; Shazeer, N.; Chen, Z. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv* **2020**, arXiv:2006.16668. [\[CrossRef\]](#)
- Fedus, W.; Zoph, B.; Shazeer, N. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.* **2022**, *23*, 1–39.
- Nguyen, V.-K.; Pham, D.-H.; Nguyen, H.-S.; Thi Nguyen, C.-V.; Le, H.-Q.; Le, D.-T. Multi-modal Adaptive Mixture of Experts for Cold-start Recommendation. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, Seoul, Republic of Korea, 10–14 November 2025; pp. 5053–5057.
- Zhou, Y.; Lei, T.; Liu, H.; Du, N.; Huang, Y.; Zhao, V.; Dai, A.M.; Le, Q.V.; Laudon, J. Mixture-of-experts with expert choice routing. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 7103–7114.
- Cheng, A.; Duan, S.; Li, S.; Yin, C.; Cheng, M.; Ping, H.; Chattopadhyay, T.; Thomopoulos, S.I.; Nazarian, S.; Thompson, P. ERMoe: Eigen-Reparameterized Mixture-of-Experts for Stable Routing and Interpretable Specialization. *arXiv* **2025**, arXiv:2511.10971.
- Abdollahpouri, H. Popularity Bias in Recommendation: A Multi-Stakeholder Perspective. Ph.D. Thesis, University of Colorado at Boulder, Boulder, CO, USA, 2020.
- Qin, J. A survey of long-tail item recommendation methods. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 7536316. [\[CrossRef\]](#)
- Zhuang, Y.; Shen, Y.; Bian, Y.; Su, Q.; Ji, S.; Shi, Y.; Miao, F. LD-MoLE: Learnable Dynamic Routing for Mixture of LoRA Experts. *arXiv* **2025**, arXiv:2509.25684.
- Quzhe, H.; Zhenwei, A.; Zhuang, N. Harder tasks need more experts: Dynamic routing in MoE models. *arXiv* **2024**, arXiv:2403.07652. [\[CrossRef\]](#)
- Xu, W.; Wu, Q.; Liang, Z.; Han, J.; Ning, X.; Shi, Y.; Lin, W.; Zhang, Y. SLMRec: Distilling Large Language Models into Small for Sequential Recommendation. In *Proceedings of the Thirteenth International Conference on Learning Representations*, Singapore, 24–28 April 2025.
- Huang, Z.; Chen, M.; Zheng, S. Dynamic Mixture of Experts for Adaptive Computation in Character-Level Transformers. *Information* **2025**, *16*, 483. [\[CrossRef\]](#)
- Xu, J.; Sun, L.; Zhao, D. MoME: Mixture-of-masked-experts for efficient multi-task recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Washington DC, USA, 14–18 July 2024; pp. 2527–2531.
- Shalaby, W.; Oh, S.; Afsharinejad, A.; Kumar, S.; Cui, X. M2TRec: Metadata-aware Multi-task Transformer for Large-scale and Cold-start Free Session-based Recommendations. In *Proceedings of the 16th ACM Conference on Recommender Systems (RecSys 2022)*, Seattle, WA, USA, 18–23 September 2022; pp. 573–578. [\[CrossRef\]](#)
- Peng, B.; Parthasarathy, S.; Ning, X. Intention enhanced mixed attentive model for session-based recommendation. *Data Min. Knowl. Discov.* **2024**, *38*, 2032–2061. [\[CrossRef\]](#)
- Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; Tikk, D. Session-based recommendations with recurrent neural networks. *arXiv* **2015**, arXiv:1511.06939.

26. 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 Advances in Neural Information Processing Systems 30 (NeurIPS 2017), Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
27. Yuan, J.; Song, Z.; Sun, M.; Wang, X.; Zhao, W.X. Dual Sparse Attention Network For Session-based Recommendation. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 4635–4643. [[CrossRef](#)]
28. Kang, W.-C.; McAuley, J. Self-attentive sequential recommendation. In Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM), Singapore, 17–20 November 2018; pp. 197–206.
29. Geng, S.; Liu, S.; Fu, Z.; Ge, Y.; Zhang, Y. Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt & Predict Paradigm (P5). In *Proceedings of the 16th ACM Conference on Recommender Systems*; Association for Computing Machinery: New York, NY, USA, 2022.
30. 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 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 2019; pp. 1441–1450.
31. Wang, X.; Qin, K.; Zhang, D.; Xie, X.; Liang, S. Bid Farewell to Seesaw: Towards Accurate Long-tail Session-based Recommendation via Dual Constraints of Hybrid Intents. *arXiv* **2025**, arXiv:2511.08378.
32. Du, N.; Huang, Y.; Dai, A.M.; Tong, S.; Lepikhin, D.; Xu, Y.; Krikun, M.; Zhou, Y.; Yu, A.W.; Firat, O. Glam: Efficient scaling of language models with mixture-of-experts. In Proceedings of the International Conference on Machine Learning, Baltimore, MD, USA, 17–23 July 2022; pp. 5547–5569.
33. Do, G.; Le, H.; Tran, T. SimSMoE: Solving Representational Collapse via Similarity Measure. *arXiv* **2024**, arXiv:2406.15883.
34. Pan, Q.; Wang, H.; An, G.; Zhang, L.; Guo, W.; Liu, Y. Revisiting scalable sequential recommendation with Multi-Embedding Approach and Mixture-of-Experts. *arXiv* **2025**, arXiv:2510.25285. [[CrossRef](#)]
35. Kim, K.; Hyun, D.; Yun, S.; Park, C. MELT: Mutual Enhancement of Long-Tailed User and Item for Sequential Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*; Association for Computing Machinery: New York, NY, USA, 2023.
36. Chen, J.; Wu, W.; Shi, L.; Zheng, W.; He, L. Long-tail session-based recommendation from calibration. *Appl. Intell.* **2023**, *53*, 4685–4702. [[CrossRef](#)]
37. Addula, S.R.; Sajja, G.S. Automated Machine Learning to Streamline Data-Driven Industrial Application Development. In Proceedings of the 2024 Second International Conference on Computational and Characterization Techniques in Engineering & Sciences (IC3TES), Lucknow, India, 15–16 November 2024; pp. 1–4.
38. Ben-Shimon, D.; Tsikinovsky, A.; Friedmann, M.; Shapira, B.; Rokach, L.; Hoerle, J. RecSys Challenge 2015 and the YOOCHOOSE Dataset. In Proceedings of the 9th ACM Conference on Recommender Systems, Vienna, Austria, 16–20 September 2015; pp. 357–358.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.