

Enhancing Long-Tail Exposure in Session-Based Recommendation via Tail-Aware Training

Oturum Tabanlı Öneri Sistemlerinde Uzun Kuyruk Görünürlüğünün Kuyruk-Farkındalıklı Eğitim Yoluyla Artırılması

Ramazan Eşmeli ^{1*} 

¹ Van Yüzüncü Yıl Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Van, Türkiye

Abstract

Recommender systems often concentrate exposure on popular items, limiting catalogue utilization and leaving long-tail items underexposed. This study evaluates TAILPromote under a leakage-controlled temporal protocol in which all eligibility, vocabulary, popularity, and long-tail definitions are derived from training data alone. TAILPromote combines a SASRec backbone with a gated tail-specific residual expert, popularity-aware logit adjustment, and catalogue-coverage regularization. Across five seeds on YooChoose 1/64, it attains MRR@20 of 33.04% $\pm$ 0.12, Coverage@20 of 89.93% $\pm$ 0.42, and LTP@20 of 39.63% $\pm$ 0.51. Paired hierarchical-bootstrap tests with Holm correction show lower accuracy than NISER+ but higher accuracy than SASRec and TailNet, together with significantly greater long-tail exposure than all three baselines. Factorial analyses identify logit adjustment as the main exposure contributor, with a smaller significant effect from Tail Expert gating. These results show that TAILPromote provides a controllable accuracy-exposure trade-off. Further evaluation on additional datasets and online settings is needed to assess generalizability and user-level utility.

Keywords: Long-tail Item Coverage, Popularity Bias, Accuracy-Diversity Trade-off

Öz

Öneri sistemleri, görünürlüğü çoğunlukla popüler öğeler üzerinde yoğunlaştırarak katalog kullanımını sınırlandırmakta ve uzun kuyruk öğelerini yeterince görünür kılmamaktadır. Bu çalışmada TAILPromote, uygunluk, sözlük, popülerlik ve uzun kuyruk tanımlarının yalnızca eğitim verisinden türetildiği veri sızıntısı kontrollü zamansal bir protokol altında değerlendirilmiştir. TAILPromote, SASRec omurgasını kapalı uzun kuyruk odaklı artık bir uzman, popülerlik duyarlı logit düzeltmesi ve katalog kapsamı düzenleştirmesi ile birleştirmektedir. YooChoose 1/64 veri kümesinde beş farklı tohum üzerinden model, MRR@20 için %33.04 $\pm$ 0.12, Coverage@20 için %89.93 $\pm$ 0.42 ve LTP@20 için %39.63 $\pm$ 0.51 değerlerine ulaşmıştır. Holm düzeltmeli eşleştirilmiş hiyerarşik bootstrap testleri, TAILPromote'un NISER+'tan daha düşük, SASRec ve TailNet'ten ise daha yüksek doğruluk sağladığını ve üç temel modele kıyasla anlamlı ölçüde daha fazla uzun kuyruk maruziyeti sunduğunu göstermektedir. Faktöriyel analizler, maruziyet artışının temel olarak logit düzeltmesinden kaynaklandığını, Tail Expert kapılamasının ise daha küçük fakat anlamlı bir katkı sağladığını göstermektedir. Sonuçlar, TAILPromote'un kontrol edilebilir bir doğruluk-maruziyet dengesi sunduğunu ortaya koymaktadır. Bulguların genellenebilirliği ve kullanıcı düzeyindeki etkilerinin değerlendirilmesi için ek veri kümeleri ve çevrim içi çalışmalar gereklidir.

Anahtar Kelimeler: Uzun Kuyruk Öğe Kapsamı, Popülerlik Yanlılığı, Doğruluk-Çeşitlilik Dengesi

* Sorumlu yazar e-posta (Corresponding e-mail): ramazanesmeli@yyu.edu.tr

Received (Geliş Tarihi):03.03.2026, Accepted (Kabul Tarihi): 27.08.2026

1. Introduction

Recommender systems (RSs) play a central role in helping users navigate the growing volume of online information [1, 2]. However, recommendation models frequently exhibit popularity bias [3, 4], assigning disproportionate exposure to already popular items while niche or long-tail content remains under-recommended [5, 6]. In session-based recommendation, where models are primarily optimized for accurate next-item prediction [7, 8], this imbalance can limit catalogue exploration and reduce opportunities for users to discover less popular items.

Existing approaches to long-tail recommendation include data augmentation, representation normalization, group-level score calibration, and post-hoc re-ranking [9, 10, 11, 12]. Although these strategies address popularity bias at different stages of the recommendation pipeline, an important question remains as to whether candidate-level tail specialization can be combined with training-time popularity control to achieve a measurable accuracy–exposure trade-off within a single framework.

To address this question, we propose TAILPromote, a tail-aware session-based recommendation framework built on a SASRec backbone whose output layer is tied to the item embeddings. TAILPromote combines a Tail-Expert Gating Mechanism, popularity-aware logit adjustment, and catalogue coverage regularization, introducing tail-awareness directly into model training rather than relying on a separate post-hoc re-ranking stage.

A key design choice in TAILPromote is to restrict the Tail Expert explicitly to long-tail candidates, giving it an identifiable role at the candidate level. We additionally compare this constrained design with an unconstrained two-expert reference configuration, allowing the effect of tail-specific specialization to be evaluated empirically rather than assumed.

We evaluate TAILPromote on the YooChoose dataset. The main contributions of this study are as follows:

1. We introduce a tail-specialized expert mechanism within a Transformer-based session recommender, with routing supervised by the popularity class of the next item.
2. We integrate popularity-aware logit adjustment and catalogue coverage regularization into the training objective, promoting long-tail exposure without requiring post-hoc re-ranking.
3. We establish a leakage-controlled temporal evaluation protocol for YooChoose 1/64, with training-only estimation of popularity-related quantities, held-out model selection, five fixed random seeds, and reproducible paired evaluation.
4. We conduct a complete factorial analysis over Tail Expert gating, tail-only logit promotion, and popularity-aware logit adjustment, complemented by paired statistical tests and separate sensitivity analysis of the catalogue coverage regularizer.

Across five random seeds, TAILPromote achieves an MRR@20 of $33.04\% \pm 0.12$, Coverage@20 of $89.93\% \pm 0.42$, and LTP@20 of $39.63\% \pm 0.51$, increasing long-tail exposure from $18.00\% \pm 0.65$ for the SASRec backbone while attaining the widest catalogue coverage among the evaluated models. NISER+ achieves the highest MRR@20 at $33.94\% \pm 0.14$, whereas TAILPromote outperforms SASRec and TailNet in MRR@20 and surpasses all evaluated baselines in long-tail exposure, catalogue coverage, and deep-tail exposure. Overall, these results show that TAILPromote substantially increases long-tail exposure and catalogue utilization while maintaining competitive ranking accuracy, providing a measurable and controllable accuracy–exposure trade-off.

The remainder of this article is organized as follows. Section 2 reviews related work on session-based recommendation and long-tail mitigation. Section 3 presents the TAILPromote framework and experimental methodology, including the leakage-controlled temporal evaluation protocol, baselines, hyperparameters, and evaluation metrics. Section 4 presents the primary results, component ablation,

popularity-group, sensitivity, and gate analyses, together with the study limitations. Section 5 concludes the article and outlines directions for future work.

2. Related Work

Session-based recommender systems (SBRs) aim to predict the next item using only the behavior observed within a short, self-contained session [13]. Unlike traditional recommenders that rely on long user histories, SBRs must operate under limited context and data sparsity while modeling complex item-transition dynamics. Early work leveraged recurrent neural networks (RNNs), notably LSTMs, to capture sequential dependencies; subsequent studies incorporated attention mechanisms and distributional item representations (e.g., Word2Vec) to further improve performance [8]. More recent research integrates graph neural networks (GNNs) to model transitions both within and across sessions [14, 15]. These GNN-based approaches account for heterogeneous user behaviors, employ contrastive objectives, and fuse local session-level embeddings with global item-transition signals [16, 17]. Additionally, methods that couple global and local graph structures, such as fusing a weighted global item graph with a local session graph, yield further accuracy gains [17]. In general, the field has progressed from RNN-based architectures to sophisticated GNN formulations that better capture the nuances of session-based recommendation.

Despite substantial progress, SBRs often suffer from the long-tail problem: a small fraction of popular items (the “short head”) attracts most recommendations and interactions, while the majority of items in the “long tail” remain underexposed. Liu and Zheng [10] were among the first to address this in session-based recommendation by partitioning items into short-head (popular) and long-tail (niche) groups. Their architecture, TailNet, calibrates the influence of each group to curb overreliance on popular products while maintaining acceptable accuracy. Other studies, such as Shafiloo et al. [18], tailored recommendations based on user tenure, showing that experienced users are generally more receptive to niche content, while Peng and Zhou [9] proposed a post-processing framework to re-rank session-based lists and promote long-tail items.

In conversational recommender systems, Zhao et al. [19] introduced LOT-CRS, which augments data for niche products and applies label smoothing to mitigate sparsity. Likewise, Yang et al. [11] presented LOAM, a data-augmentation approach that preserves natural user transitions while increasing tail exposure. Other strategies include leveraging deep architectures for cold start [20], mining item co-occurrence for complementary recommendations [21], and post-training embedding enhancement for long-tail recommendation [22]. Frameworks developed by [23] and [24] employ transfer learning and full-index retrieval to improve long-tail visibility. In sequential and service recommendation, in [5] and [6] contextual cues and cross-view representations were used to correct popularity bias. Multi-objective optimization [25] further helps reconcile accuracy with diversity.

In contrast to methods that operate solely via data augmentation [11] or rely on post-hoc re-ranking [9], TAILPromote embeds long-tail promotion directly into the representation learning and scoring phases. Whereas TailNet [10] rectifies full-catalogue scores with a dual-pathway head/tail preference factor that is applied uniformly to every item of a group, and NISER [12] attenuates popularity implicitly by normalizing item and session representations, TAILPromote integrates a Transformer-based (SASRec) [26] backbone with a Tail Expert whose candidate support is explicitly restricted to training-defined tail items. Three properties distinguish it from both: the expert is identifiable at the candidate level rather than only at the representation level; routing is supervised by the popularity class of the next item rather than learned implicitly; and the popularity-aware loss and the batch-level coverage regularizer remain separable from the architecture. The independent effects of Tail Expert gating and popularity-aware logit adjustment are evaluated within the component factorial in Section 4.3, alongside tail-only logit promotion, while the catalogue coverage regularizer is examined separately through sensitivity analysis.

3. Materials and Methods

3.1. Framework and Notation

The notation used in the TAILPromote architecture is summarized in Table 1. The model represents a sequence of user interactions as an input session, which is transformed into an embedding matrix and processed through a Transformer backbone to obtain a session representation. The Main Head scores this representation against the shared item embedding matrix, while the Tail Expert acts as a residual adapter restricted to training-defined tail candidates. The Tail Expert correction is read out through the same item embeddings and added only to the tail entries of the Main Head scores, with its contribution controlled by a scalar gating network.

Table 1. Summary of TAILPromote Notations

Notation	Definition and Function
$S = \{i_1, i_2, \dots, i_L\}$	Input session sequence of item indices with a maximum length L .
$E \in R^{L \times d}$	Input embedding matrix combining item and learnable positional embeddings.
d	Dimensionality of the embeddings, set to 200.
$h_L \in R^d$	Final session representation (hidden state) produced by the Transformer encoder.
$g \in [0,1]$	Learnable gating scalar that controls the contribution of the Tail Expert correction.
σ	The Sigmoid activation function used within the gating network.
W_{g1}, W_{g2}	Specific learnable weight matrices used to calculate the gating score g
L	The Logit Adjustment Loss function used to penalize over-confidence on popular items.
p_y	Prior popularity probability of the target item used to adjust the loss margin.
T	Tail candidate set, defined from the filtered training counts alone.
1T	Indicator of the tail candidate set, confining the correction to T.
E	Item embedding matrix shared by the encoder, Main Head scoring function, and Tail Expert readout.
a(h)	Tail Expert adapter output, a correction in the session representation space.
zmix	Final scores: Main Head scores plus the gated correction on the tail support.
ty	Routing target, equal to 1 when the next item lies in T and 0 otherwise.
w +	Positive class weight of the gate supervision term.
$\bar{q}$	Batch-averaged predictive distribution entering the coverage penalty.
$\lambda_g, \lambda_t, \lambda_{cov}$	Weights of the gate-supervision, tail-ranking and catalogue-coverage terms.

To increase the visibility of tail items, TAILPromote builds upon a Transformer-based session encoder (Figure 1) whose output layer is tied to the item embeddings, and adds three mechanisms that act during training rather than after ranking: a Tail Expert adapter read out through those same embeddings and confined to the tail support, under a supervised routing gate; a popularity-aware logit adjustment loss; and a batch-level catalogue coverage regularizer. A fourth mechanism, tail-only logit promotion, is retained as a switch of the component factorial in Section 4.3; it did not demonstrate an independent benefit and is not part of the primary configuration.

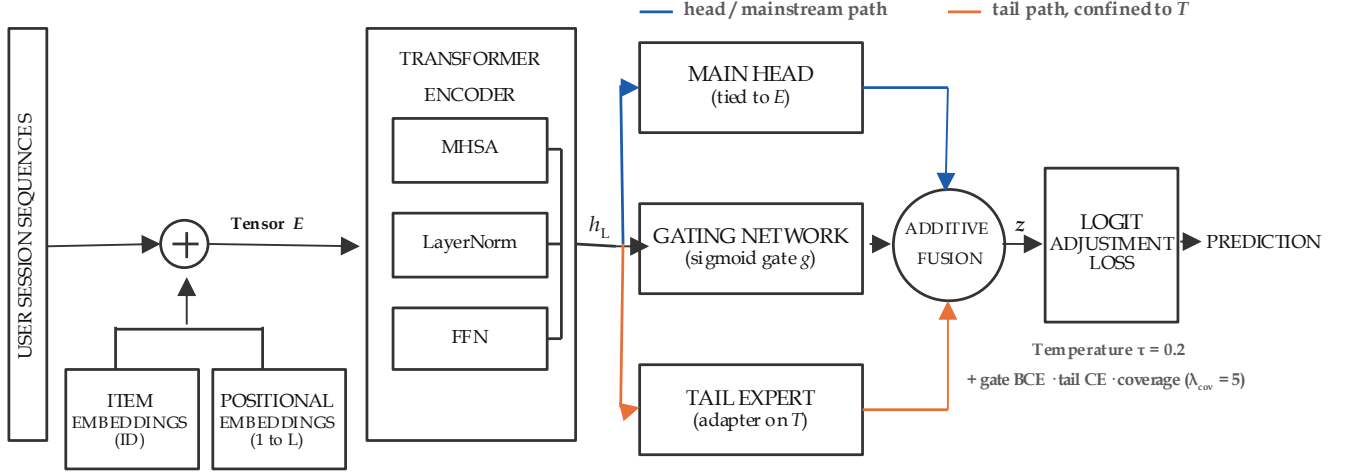


Figure 1. Schematic overview of the TAILPromote framework. The architecture consists of (1) a SASRec backbone that processes item and positional embeddings under causal and padding masks to produce the session representation; (2) a Main Head that scores that representation against the item embedding matrix used by the encoder, so that the output layer introduces no additional parameters; (3) a Tail Expert implemented as a residual adapter, read out through the same embeddings and confined to the training-defined tail support, whose contribution is scaled by a learnable sigmoid gate under class-balanced routing supervision; and (4) a popularity-aware logit adjustment loss together with a batch-level catalogue coverage regularizer. Because the Tail Expert correction is additive and restricted to the tail support, scores of head items are left unchanged.

3.1.1. Backbone Encoder (SASRec)

First, we construct the input embedding matrix $E \in R^{L \times d}$ by summing item embeddings and learnable positional embeddings. Unlike RNN-based approaches, we utilize the SASRec [26] architecture as the backbone. The sequence is processed by stacked Transformer blocks, each consisting of Multi-Head Self-Attention (MHSA) and Feed-Forward Networks (FFN) (Figure 1). The final hidden state of the last item denoted as $h_L \in R^d$ serves as the comprehensive session representation capturing the user's current intent.

3.1.2. Tail-Expert Gating Mechanism

To introduce explicit tail-specific specialization, we augment the SASRec session representation with a gated Tail Expert. The session representation is processed through a Main Head and a tail-specific correction pathway:

Main Head: The session representation h is scored against the shared item embedding matrix E to produce the base item scores, as defined in Equation (3).

Tail Expert: A specialized non-linear Multi-Layer Perceptron (MLP) designed to detect niche signals as seen in Equation (1).

$$z_{tail} = W_{t2} \left(\text{LeakyReLU}(W_{t1} h_L) \right). \quad (1)$$

To dynamically balance these signals, a learnable Gating Network computes a scalar weight $g \in [0,1]$ based on the session context as given in Equation (2).

$$g = \sigma \left(W_{(g2)\text{ReLU}}(W_{(g1)} h_L) \right) \quad (2)$$

where σ denotes the sigmoid function and ReLU is the Rectified Linear Unit activation. The two pathways are not combined as unconstrained logit vectors. The Main Head scores the session representation h against the item embedding matrix E that the encoder itself uses, as given in Equation (3), so that the output layer introduces no parameters of its own. The Tail Expert is an adapter rather than a second scorer: it maps h to a correction in the same representation space, which is read out through the same embeddings and confined to the training-defined tail candidate set $T \subset V$ by the indicator 1_T , and the scalar gate g weights that correction, as given in Equation (4). The Tail Expert directly modifies only tail-item scores, while head-item logits remain unchanged; because the correction is expressed in the shared item space, the expert reuses the catalogue representation instead of learning a second copy of it. The gate g is the weight the model places on the tail correction, and it is this quantity that is analysed in Section 4.4.

$$z_{\text{main}} = Eh, g = \sigma(w^\top h) \quad (3)$$

$$z_{\text{mix}} = z_{\text{main}} + g(Ea(h)) \odot 1_T \quad (4)$$

Here $a(\cdot)$ denotes the adapter and $\odot$ the elementwise product. Three auxiliary objectives render the intended behaviour explicit rather than incidental. The routing target t_y equals 1 when the next item lies in T and 0 otherwise, and a class-balanced binary cross-entropy with positive class weight w^+ supervises the gate directly, as given in Equation (5); supervising the gate matters because a routing weight that is trained only through the ranking loss has no reason to align with tail membership. The Tail Expert additionally receives a cross-entropy evaluated over T on those examples whose target is a tail item, as given in Equation (6). Finally, catalogue coverage is a property of a set of recommendations rather than of any single one, and a per-example ranking loss is indifferent to whether every session is served the same items; Equation (7) therefore penalizes the normalized negentropy of the batch-averaged predictive distribution $\bar{q}$, which is zero when a batch spreads its mass uniformly over the catalogue and approaches one when it collapses onto a single item. The penalty constrains only the batch average, leaving individual predictions as peaked as the ranking loss requires. The complete training objective is the weighted sum in Equation (8).

$$L_{\text{gate}} = -[w^+ t_y \log g + (1 - t_y) \log (1 - g)] \quad (5)$$

$$L_{\text{tail}} = -1[y \in T] \log q_{\text{tail},y} \quad (6)$$

$$L_{\text{cov}} = 1 - H(\bar{q}) / \log|V|, \bar{q} = 1/B \cdot \Sigma q_b \quad (7)$$

$$L_{\text{total}} = L_{\text{rank}} + \lambda_g L_{\text{gate}} + \lambda_t L_{\text{tail}} + \lambda_{\text{cov}} L_{\text{cov}} \quad (8)$$

The positive class weight w^+ is the ratio of head-target to tail-target counts on the training partition. The auxiliary weights (λ_g, λ_t) were selected on the validation partition alone, over the pairs (0.05, 0.25), (0.10, 0.50) and (0.20, 1.00), with the ratio between them held fixed at 1:5; the pair (0.05, 0.25) was retained. The searches over the auxiliary weights, the temperature and the coverage weight were conducted sequentially, each on the validation partition alone, and none consulted the test partition. Section 4.5 notes the scope of this search.

Equation (9) states the alternative combination in which two unconstrained expert logit vectors are mixed by the same scalar weight. This configuration is carried as a paired reference in Sections 4.1 and 4.4; the model presented here uses Equation (4).

$$z_{\text{combined}} = (1 - g) \cdot z_{\text{main}} + g \cdot z_{\text{tail}} \quad (9)$$

3.1.3. Logit-Level Tail Promotion

As an additional mechanism evaluated in the component factorial, a learnable additive boost can be applied directly to the logits of tail items. Let 1_T denote the binary tail mask defined in Equation (4), and let b be a learnable bias vector initialized to zero. The resulting adjusted scores are defined in Equation (10). This mechanism constitutes the promotion switch P in the $2 \times 2 \times 2$ factorial analysis of Section 4.3. Because it did not demonstrate an independent benefit, it is not included in the primary TAILPromote configuration; its empirical effect is reported in Section 4.3.

$$z_{final} = z_{combined} + (\beta \odot M_{tail}) \quad (10)$$

3.1.4. Training Objective (Logit Adjustment Loss)

To mitigate popularity bias during training, we use popularity-aware logit-adjusted cross-entropy as the ranking term L_{rank} in Equation (8). The loss incorporates training-derived item popularity priors into the ranking objective, as defined in Equation (11).

$$\mathcal{L} = -\log \frac{\exp(z_y + \tau \cdot \log p(y))}{\sum_{j \in I} \exp(z_j + \tau \cdot \log p(j))} \quad (11)$$

where τ is the logit-adjustment temperature controlling the strength of the popularity correction. Its value is selected using the held-out validation partition, as described in Section 3.3.

3.2. Dataset and Leakage-Controlled Temporal Evaluation

We conducted our evaluation on the YooChoose dataset [27], derived from the RecSys Challenge 2015, which contains session-based interaction data from an e-commerce platform. Following prior session-based recommendation studies, we utilize the most recent $1/64$ fraction of the training sequences. This subset preserves the temporal characteristics of user behavior while reducing computational overhead, which is particularly important for evaluating long-tail recommendation performance.

We follow preprocessing steps:

- **Temporal Split:** Complete sessions are assigned to partitions according to their final UTC timestamp before any sampling or filtering is applied. Sessions ending within the final 24 hours constitute the test partition, those ending in the preceding 24 hours constitute the validation partition, and all earlier sessions form the training pool, from which the most recent $1/64$ of complete sessions is retained. The three sets of session identifiers are disjointed by construction. All subsequent filtering, item indexing, popularity estimation, and long-tail definitions are derived exclusively from the training partition.
- **Filtering:** An iterative fixed-point filter is applied to the retained training sessions only. Items with fewer than five training interactions are removed, sessions retaining fewer than two interactions are then removed, and these two steps are repeated until neither set changes further. Validation and test interactions whose items fall outside the resulting training vocabulary are subsequently removed, as are evaluation sessions retaining fewer than two eligible events. No validation or test interaction contributes to any eligibility decision.
- **Sequence Handling:** Each eligible session yields a single next-item prediction example, with the final event used as the target and the preceding events forming the input sequence. Input prefixes longer than 18 events are truncated to their 18 most recent events, so that no session is discarded because of its length. Padding is excluded from model computations where applicable, candidate logits, top-20 recommendation lists, and all reported metrics.

- Training-Derived Quantities: Item counts, popularity priors, the head/tail mask, popularity deciles, and the head, middle-tail, and deep-tail groups are estimated exclusively from the filtered training interactions and then applied unchanged to the validation and test partitions.
- Indexing: Item identifiers are re-indexed as contiguous integers derived exclusively from the training vocabulary, with indices assigned from 1 upward in ascending order of the original item identifiers. Index 0 is reserved exclusively for padding and is excluded from the candidate catalogue.

The statistics of the pre-processed dataset are summarized in Table 2.

Table 2. Statistics of the YooChoose 1/64 Dataset

Feature	Value
Training sessions / examples	106,013
Validation sessions / examples	12,032
Test sessions / examples	14,900
Eligible catalogue items (training-frozen)	8,367
Training interactions	433,503
Item-frequency Gini	0.7593
Session-item sparsity	99.95%

3.3. Baselines, Hyperparameters, and Metrics

All models were implemented in PyTorch and executed on a single workstation equipped with an NVIDIA T1000 GPU (4 GB VRAM) under CUDA 12.8. The same device, data pipeline, and evaluation code were used for all models.

Three baselines are compared with TAILPromote under this shared protocol. SASRec [26] serves as the Transformer backbone of TAILPromote and scores candidate items using the item-embedding matrix shared with the sequence encoder. NISER+ [12] normalizes graph-derived item and session representations; the NISER+ configuration including positional embeddings and input-embedding dropout is used consistently throughout the experiments. TailNet [10] calibrates head and tail item groups using a session-level preference factor. In our controlled implementation, these factors are applied to the softmax-normalized catalogue scores rather than directly to raw logits, preserving their intended multiplicative interpretation. All reported TailNet results therefore correspond to this implementation under the same data and evaluation protocol used for the other models. The unconstrained two-expert mixture defined in Equation (9) is additionally included as a paired reference configuration.

3.3.1. General Hyperparameters

To maintain a consistent experimental protocol, shared training settings were held fixed across models where applicable:

- Embedding Dimension (d): Set to 200.
- Batch Size: Set to 2048 for all reported configurations.
- Optimization: Models were trained using Adam with $\beta_1=0.9$ and $\beta_2=0.999$, together with a OneCycleLR scheduler with a maximum learning rate of 0.005. Each configuration was trained for exactly eight epochs.
- Maximum Sequence Length: Set to 18 interactions, consistent with the sequence-handling procedure described in Section 3.2.
- Dropout: A dropout rate of 0.2 was used in architecture-specific dropout layers where applicable.
- Attention Heads: Four attention heads were used in the SASRec backbone shared by SASRec and TAILPromote.

No early stopping or metric-based checkpoint selection was used; the state after the eighth epoch was used for evaluation, as detailed below.

3.3.2. TAILPromote-Specific Parameters

TAILPromote-specific hyperparameters were selected using the held-out validation partition only, before any evaluation on the test partition was performed.

- Auxiliary-loss weights: The gate-supervision and tail-ranking weights (λ_g, λ_t) were evaluated at (0.05,0.25), (0.10,0.50), and (0.20,1.00), with their ratio fixed at 1:5. The pair (0.05,0.25) was retained based on validation performance.
- Logit-adjustment temperature and coverage weight: The reported configuration uses $\tau=0.2$ and $\lambda_{cov}=5$. A setting was considered admissible when the gate assigned greater mean weight to tail-target than to head-target sessions in every seed and its five-seed mean validation MRR@20 retained at least 95% of the corresponding SASRec validation mean (29.19%). Within this admissible set, $\tau=0.2$ and $\lambda_{cov}=5$ were selected on the validation partition as the reported operating point to represent a moderate accuracy–exposure balance. The full validation sensitivity analyses for τ and λ_{cov} , including the trade-off trajectories, are reported in Section 4.4. No test result was used during hyperparameter selection.

3.3.3. Evaluation, Repetition, and Statistical Analysis

Evaluation Protocol: All models were evaluated at $K=20$ using identical encoded examples, targets, and candidate sets. Each reported configuration was trained for exactly eight epochs. Validation metrics were recorded during training, but no early stopping or metric-based checkpoint selection was used; the model state after the eighth epoch was used for evaluation. Hyperparameter selection was performed exclusively on the validation partition. The test partition was not evaluated during training or model selection and was accessed only after the configuration, and selection rules had been fixed.

Seeds and Repetition: Random seeds were fixed at 17, 42, 123, 2026, and 3407, and every reported configuration was trained five times, once per seed. Python, NumPy, PyTorch CPU and CUDA random number generators, and the data-loading pipeline were seeded consistently. cuDNN benchmarking was disabled and deterministic attention kernels were used. All reported results are presented as the arithmetic mean and sample standard deviation over the five seeds.

Statistical Analysis: TAILPromote was compared with SASRec, NISER+, TailNet, and the paired unconstrained-mixture reference configuration on MRR@20, Coverage@20, LTP@20, and deep-tail exposure using 10,000 paired hierarchical-bootstrap replicates. Each replicate resampled seeds with replacement and then resampled matched test examples within each sampled seed, thereby incorporating both seed-level and example-level variation. Because Coverage@20 is a set-union measure, it was recomputed from the sampled top-20 recommendation lists within each replicate rather than averaged across individual examples. Two-sided percentile 95% confidence intervals and two-sided bootstrap p-values were computed, with Holm correction applied separately within each metric across the four comparisons. A difference was considered statistically significant only when the Holm-adjusted p-value was below 0.05 and the corresponding 95% confidence interval excluded zero.

3.3.4. Evaluation Metrics

To evaluate the proposed long-tail-aware recommendation framework, we use ranking-accuracy, catalogue-utilization, and exposure-oriented metrics. All metrics are reported at $K=20$.

- MRR@20 is used as the primary ranking-accuracy metric. It measures the reciprocal rank of the ground-truth next item within the top-20 recommendation list, assigning a value of zero when the target does not appear in the list. Higher MRR@20 indicates that relevant targets are ranked closer to the top of the recommendation list.
- HitRate@20 is additionally reported in the popularity-group analysis. It measures the proportion of test examples for which the ground-truth next item appears within the top-20 recommendations.
- Coverage@20 measures catalogue utilization and is defined as the proportion of eligible catalogue items that appear at least once across the top-20 recommendation lists. The denominator is the training-defined eligible catalogue described in Section 3.2. Higher Coverage@20 indicates that recommendations are distributed across a broader portion of the available catalogue.
- LTP@20 measures long-tail exposure. It is defined as the proportion of all recommended positions in the top-20 lists that are occupied by items belonging to the training-defined long-tail set. LTP@20 therefore quantifies exposure to less-popular items rather than their ranking accuracy or relevance.

Because a binary head/tail division cannot distinguish moderately infrequent items from the least-popular items, the catalogue is additionally partitioned into popularity groups using deciles of training-item frequency. Items are ordered by descending training frequency, with ties broken deterministically by item identifier. The head comprises decile 1, corresponding to the most frequent 10% of eligible catalogue items; the middle tail comprises deciles 2–5; and the deep tail comprises deciles 6–10. These groups are derived exclusively from the training partition and remain fixed across all models, validation data, and test data.

For each popularity group, we report recommendation-position exposure, target-conditioned MRR@20 and HitRate@20, and within-group catalogue coverage. Exposure denotes the proportion of top-20 recommendation positions occupied by items from the corresponding group, whereas within-group coverage denotes the proportion of items in that group that appear at least once in the recommendation lists.

Under the primary 10% head definition, LTP@20 and head exposure are complementary and sum to one. In the head-fraction sensitivity analysis, the head threshold used to define LTP@20 is recomputed at 5%, 10%, and 20%, while the head, middle-tail, and deep-tail exposure groups remain fixed according to the decile-based definition above. Consequently, LTP@20 and head exposure are complementary only under the primary 10% head setting.

4. Results and Discussion

This section compares TAILPromote with SASRec, NISER+, and TailNet on the YooChoose 1/64 sample in terms of ranking accuracy, catalogue utilization, and long-tail exposure. Ranking accuracy is assessed using MRR@20, while Coverage@20, LTP@20, and deep-tail exposure characterize how recommendation positions are distributed across the catalogue. Reported model-level values are means with sample standard deviations over five seeds, and paired model differences are accompanied by confidence intervals and Holm-adjusted significance tests.

4.1. Primary Results

Table 3 summarizes the primary test performance of all models, and Figure 2 visualizes their positions in the accuracy–exposure space. Among the primary models, NISER+ attains the highest MRR@20, whereas TAILPromote attains the highest Coverage@20 and LTP@20. The unconstrained mixture provides a separate reference point, illustrating a further increase in exposure accompanied by a substantial loss in ranking accuracy.

Table 3. Primary test results over five seeds (%), reported as mean $\pm$ sample standard deviation for seeds 17, 42, 123, 2026, and 3407. All models are evaluated on identical encoded examples, targets, and candidate sets. The final row reports the unconstrained two-expert mixture defined in Equation (9) as a paired reference configuration.

Model	MRR@20	Coverage@20	LTP@20
SASRec [26]	30.56% $\pm$ 0.60	62.34% $\pm$ 2.55	18.00% $\pm$ 0.65
NISER+ [12]	33.94% $\pm$ 0.14	87.81% $\pm$ 0.19	24.54% $\pm$ 0.08
TailNet [10]	31.05% $\pm$ 0.21	81.69% $\pm$ 0.56	24.25% $\pm$ 0.18
TAILPromote (Ours)	33.04% $\pm$ 0.12	89.93% $\pm$ 0.42	39.63% $\pm$ 0.51
TAILPromote (unconstrained mixture)	27.65% $\pm$ 0.49	79.14% $\pm$ 3.18	45.72% $\pm$ 0.58

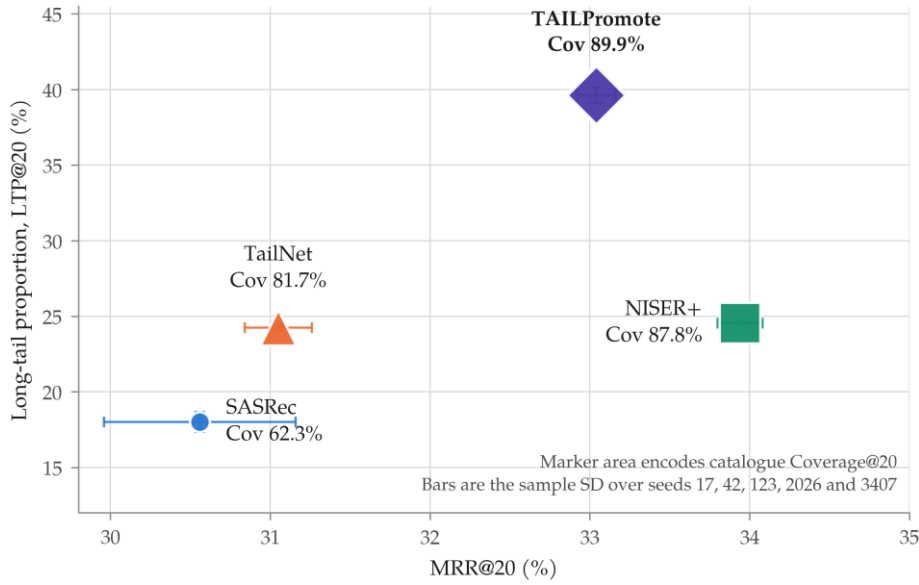


Figure 2. Accuracy-exposure trade-off over five seeds for SASRec, NISER+, TailNet, and TAILPromote under the YooChoose 1/64 protocol. The horizontal axis shows MRR@20 and the vertical axis shows LTP@20; marker area represents Coverage@20, and error bars indicate the sample standard deviation across seeds 17, 42, 123, 2026, and 3407.

SASRec attains an MRR@20 of 30.56% $\pm$ 0.60, NISER+ 33.94% $\pm$ 0.14, TailNet 31.05% $\pm$ 0.21, and TAILPromote 33.04% $\pm$ 0.12. NISER+ is therefore the most accurate model among the primary models. The paired MRR@20 difference for TAILPromote is +2.48 percentage points relative to SASRec (95% CI +1.99 to +3.02, Holm-adjusted $p < 0.0002$), -0.90 points relative to NISER+ (95% CI -1.10 to -0.70, $p < 0.0002$), and +1.99 points relative to TailNet (95% CI +1.77 to +2.19, $p < 0.0002$). Thus, TAILPromote is significantly more accurate than SASRec and TailNet, but significantly less accurate than NISER+.

The clearest differences between the models occur in long-tail exposure and catalogue coverage. TAILPromote attains an LTP@20 of 39.63% $\pm$ 0.51, compared with 18.00% $\pm$ 0.65 for SASRec, 24.54% $\pm$ 0.08 for NISER+, and 24.25% $\pm$ 0.18 for TailNet. The paired differences are +21.62 percentage points relative to SASRec (95% CI +20.94 to +22.20, Holm-adjusted $p < 0.0002$), +15.08 points relative to NISER+ (95% CI +14.63 to +15.53, $p < 0.0002$), and +15.38 points relative to TailNet (95% CI +14.97 to +15.89, $p < 0.0002$). Deep-tail exposure shows the same pattern among the primary models, reaching 12.78% for TAILPromote compared with 2.41% for SASRec, 6.01% for NISER+, and 5.39% for TailNet.

TAILPromote also attains the highest catalogue coverage among the primary models, with Coverage@20 of 89.93% $\pm$ 0.42, compared with 87.81% $\pm$ 0.19 for NISER+, 81.69% $\pm$ 0.56 for TailNet, and 62.34% $\pm$ 2.55 for SASRec. The paired Coverage@20 differences are +22.20 percentage points relative to SASRec (95% CI +20.43 to +25.30), +1.01 points relative to NISER+ (95% CI +0.47 to +1.82), and +5.62 points relative to

TailNet (95% CI +4.94 to +6.70); all remain significant after Holm correction. Section 4.2 examines whether the greater tail exposure is concentrated in the middle tail or extends to the deep tail, while distinguishing exposure from ranking accuracy.

Taken together, these results show a reproducible redistribution of recommendation exposure toward less-popular items. This redistribution should not be interpreted as a uniform improvement in relevance: Section 4.2 shows that the increase in deep-tail exposure does not translate into higher deep-tail ranking accuracy.

Table 4 reports the paired hierarchical-bootstrap comparisons between TAILPromote and each baseline, together with the unconstrained-mixture reference configuration.

Table 4. Paired hierarchical-bootstrap comparisons of TAILPromote with the three baselines and the unconstrained-mixture reference configuration over 10,000 replicates, reported in percentage points. Values show mean paired differences, 95% confidence intervals, and Holm-adjusted p-values. Positive differences favour TAILPromote.

Metric	Baseline	Mean difference	95% CI	Holm-adjusted p	Significant
MRR@20	SASRec	+2.48	[+1.99, +3.02]	< 0.0002	yes
MRR@20	NISER+	-0.90	[-1.10, -0.70]	< 0.0002	yes
MRR@20	TailNet	+1.99	[+1.77, +2.19]	< 0.0002	yes
MRR@20	TAILPromote (unconstr. mixture)	+5.39	[+4.92, +5.78]	< 0.0002	yes
Coverage@20	SASRec	+22.20	[+20.43, +25.30]	< 0.0002	yes
Coverage@20	NISER+	+1.01	[+0.47, +1.82]	< 0.0002	yes
Coverage@20	TailNet	+5.62	[+4.94, +6.70]	< 0.0002	yes
Coverage@20	TAILPromote (unconstr. mixture)	+6.94	[+5.80, +9.54]	< 0.0002	yes
LTP@20	SASRec	+21.62	[+20.94, +22.20]	< 0.0002	yes
LTP@20	NISER+	+15.08	[+14.63, +15.53]	< 0.0002	yes
LTP@20	TailNet	+15.38	[+14.97, +15.89]	< 0.0002	yes
LTP@20	TAILPromote (unconstr. mixture)	-6.09	[-6.97, -5.32]	< 0.0002	yes
Deep-tail exposure	SASRec	+10.37	[+9.94, +10.74]	< 0.0002	yes
Deep-tail exposure	NISER+	+6.77	[+6.46, +7.04]	< 0.0002	yes
Deep-tail exposure	TailNet	+7.39	[+7.09, +7.67]	< 0.0002	yes
Deep-tail exposure	TAILPromote (unconstr. mixture)	-3.57	[-4.48, -2.96]	< 0.0002	yes

The clearest differences between the models occur in long-tail exposure and catalogue coverage. TAILPromote attains an LTP@20 of 39.63% $\pm$ 0.51, compared with 18.00% $\pm$ 0.65 for SASRec, 24.54% $\pm$ 0.08 for NISER+, and 24.25% $\pm$ 0.18 for TailNet. The paired differences are +21.62 percentage points relative to SASRec (95% CI +20.94 to +22.20, Holm-adjusted $p < 0.0002$), +15.08 points relative to NISER+ (95% CI +14.63 to +15.53, $p < 0.0002$), and +15.38 points relative to TailNet (95% CI +14.97 to +15.89, $p < 0.0002$). Deep-tail exposure shows the same pattern among the primary models, reaching 12.78% for TAILPromote compared with 2.41% for SASRec, 6.01% for NISER+, and 5.39% for TailNet.

TAILPromote also attains the highest catalogue coverage among the primary models, with Coverage@20 of 89.93% $\pm$ 0.42, compared with 87.81% $\pm$ 0.19 for NISER+, 81.69% $\pm$ 0.56 for TailNet, and 62.34% $\pm$ 2.55 for SASRec. The paired Coverage@20 differences are +22.20 percentage points relative to SASRec (95% CI +20.43 to +25.30), +1.01 points relative to NISER+ (95% CI +0.47 to +1.82), and +5.62 points relative to TailNet (95% CI +4.94 to +6.70); all remain significant after Holm correction. Section 4.2 examines whether the greater tail exposure is concentrated in the middle tail or extends to the deep tail, while distinguishing exposure from ranking accuracy.

4.2. Popularity-Group and Deep-Tail Analysis

To distinguish moderately infrequent items from the least-popular items, the catalogue is partitioned into the head, middle-tail, and deep-tail groups defined in Section 3.3. Table 5 reports group-specific ranking

accuracy, exposure, and catalogue coverage, while Figure 3 shows recommendation-position exposure across the three popularity groups for all four primary models.

Table 5. Test results by training-defined popularity group over five seeds (%), reported as mean $\pm$ sample standard deviation. Targets denote the number of test examples in each group; Exposure@20 is the proportion of top-20 recommendation positions occupied by items from that group, and Coverage@20 is the proportion of items within the group that appear in at least one recommendation list.

Model	Group	Targets	MRR@20	HitRate@20	Exposure@20	Coverage@20
SASRec	head	11,455	34.21% $\pm$ 0.43	70.28% $\pm$ 0.46	82.00% $\pm$ 0.65	99.50% $\pm$ 0.26
SASRec	middle-tail	2,719	20.54% $\pm$ 1.28	41.41% $\pm$ 2.04	15.59% $\pm$ 0.46	80.14% $\pm$ 2.11
SASRec	deep-tail	726	10.38% $\pm$ 1.28	21.74% $\pm$ 1.87	2.41% $\pm$ 0.21	40.66% $\pm$ 3.44
NISER+	head	11,455	36.04% $\pm$ 0.14	72.11% $\pm$ 0.05	75.46% $\pm$ 0.08	100.00% $\pm$ 0.00
NISER+	middle-tail	2,719	27.76% $\pm$ 0.25	54.38% $\pm$ 0.22	18.53% $\pm$ 0.10	96.64% $\pm$ 0.10
NISER+	deep-tail	726	23.94% $\pm$ 0.30	45.90% $\pm$ 0.42	6.01% $\pm$ 0.03	78.31% $\pm$ 0.40
TailNet	head	11,455	34.79% $\pm$ 0.23	70.56% $\pm$ 0.14	75.75% $\pm$ 0.18	100.00% $\pm$ 0.00
TailNet	middle-tail	2,719	20.86% $\pm$ 0.33	42.63% $\pm$ 0.80	18.85% $\pm$ 0.20	93.60% $\pm$ 0.41
TailNet	deep-tail	726	10.33% $\pm$ 0.28	19.09% $\pm$ 0.66	5.39% $\pm$ 0.06	68.50% $\pm$ 0.87
TAILPromote	head	11,455	34.32% $\pm$ 0.05	69.56% $\pm$ 0.36	60.37% $\pm$ 0.51	99.95% $\pm$ 0.07
TAILPromote	middle-tail	2,719	30.30% $\pm$ 0.41	55.20% $\pm$ 0.72	26.84% $\pm$ 0.27	96.92% $\pm$ 0.12
TAILPromote	deep-tail	726	23.20% $\pm$ 0.51	39.70% $\pm$ 1.27	12.78% $\pm$ 0.38	82.33% $\pm$ 0.76

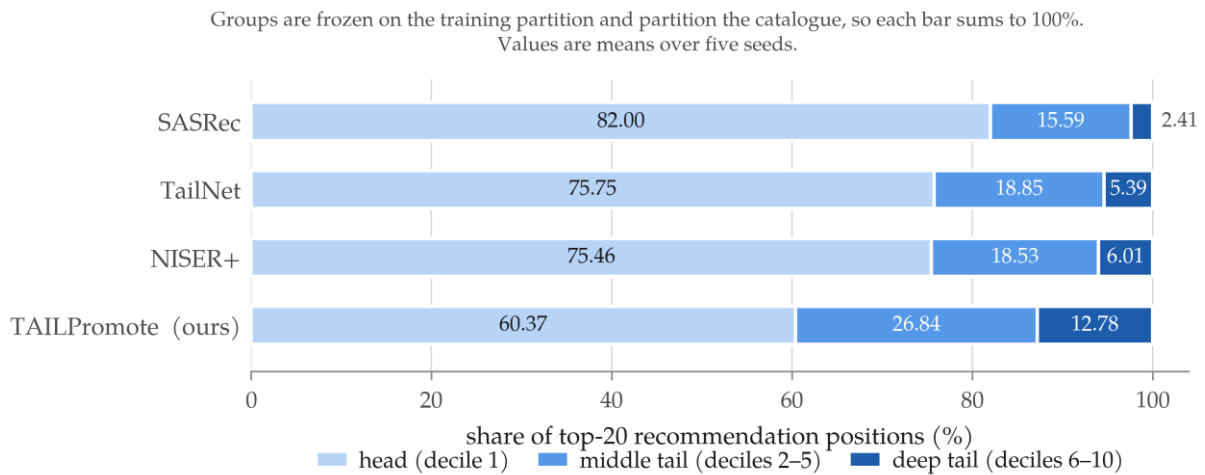


Figure 3. Recommendation-position exposure by training-defined popularity group for SASRec, NISER+, TailNet, and TAILPromote. The head, middle-tail, and deep-tail groups partition all top-20 recommendation positions, so each model’s shares sum to 100%. Values are five-seed means plotted directly without normalization or rescaling.

The group-level results show that TAILPromote redistributes recommendation exposure substantially beyond the head. Head exposure decreases to 60.37% $\pm$ 0.51, compared with 82.00% $\pm$ 0.65 for SASRec, 75.46% $\pm$ 0.08 for NISER+, and 75.75% $\pm$ 0.18 for TailNet. Correspondingly, TAILPromote attains the highest middle-tail exposure (26.84% $\pm$ 0.27) and deep-tail exposure (12.78% $\pm$ 0.38) among the primary models. It also achieves the highest within-group catalogue coverage in the middle tail (96.92% $\pm$ 0.12) and deep tail (82.33% $\pm$ 0.76).

The ranking results, however, differ by popularity group. For middle-tail targets, TAILPromote attains the highest MRR@20 (30.30% $\pm$ 0.41) and HitRate@20 (55.20% $\pm$ 0.72) among the evaluated models. For deep-tail targets, NISER+ attains the highest ranking accuracy, with MRR@20 of 23.94% $\pm$ 0.30 and HitRate@20 of 45.90% $\pm$ 0.42, compared with 23.20% $\pm$ 0.51 and 39.70% $\pm$ 1.27 for TAILPromote. Thus, TAILPromote’s strongest deep-tail result is increased exposure and catalogue coverage rather than improved deep-tail ranking accuracy.

4.3. Component Ablation

To quantify the contribution of the individual tail-aware mechanisms, we evaluate the complete $2 \times 2 \times 2$ factorial over Tail Expert gating (G), tail-only logit promotion (P), and popularity-aware logit adjustment (A) across all five seeds. The catalogue coverage regularizer is held outside this factorial and is examined separately through the sensitivity analysis in Section 4.4. Table 6 reports all eight factorial configurations on the test partition, and Figure 4 shows their positions in the accuracy–exposure space.

Table 6. Complete $2 \times 2 \times 2$ factorial over Tail Expert gating (G), tail-only logit promotion (P), and logit adjustment (A) on the test partition over five seeds (%), reported as mean $\pm$ sample standard deviation. The G=0,P=0,A=0 row denotes the all-off factorial configuration and should not be interpreted as the standalone SASRec baseline reported in Table 3, because the remaining TAILPromote training framework is retained. The final row reports a separate independent-Main-Head control and is not a factorial cell.

G	P	A	MRR@20	HitRate@20	Coverage@20	LTP@20	Deep-tail exp.
off	Off	off	33.10% $\pm$ 0.16	65.22% $\pm$ 0.24	80.08% $\pm$ 0.83	28.95% $\pm$ 0.49	6.63% $\pm$ 0.16
off	Off	on	33.48% $\pm$ 0.14	65.43% $\pm$ 0.20	87.27% $\pm$ 0.86	36.47% $\pm$ 0.87	12.13% $\pm$ 0.72
off	On	off	32.99% $\pm$ 0.18	65.03% $\pm$ 0.44	79.23% $\pm$ 1.76	28.58% $\pm$ 0.67	6.42% $\pm$ 0.50
off	On	on	33.35% $\pm$ 0.09	65.22% $\pm$ 0.25	87.15% $\pm$ 0.98	36.41% $\pm$ 0.99	12.04% $\pm$ 0.82
on	Off	off	33.12% $\pm$ 0.06	65.65% $\pm$ 0.19	87.28% $\pm$ 0.58	33.22% $\pm$ 0.32	8.72% $\pm$ 0.17
on	Off	on	33.04% $\pm$ 0.12	65.49% $\pm$ 0.35	89.93% $\pm$ 0.42	39.63% $\pm$ 0.51	12.78% $\pm$ 0.38
on	On	off	33.23% $\pm$ 0.12	65.93% $\pm$ 0.20	87.40% $\pm$ 0.16	33.63% $\pm$ 0.51	8.86% $\pm$ 0.27
on	On	on	33.08% $\pm$ 0.05	65.50% $\pm$ 0.35	89.96% $\pm$ 0.10	39.41% $\pm$ 0.53	12.62% $\pm$ 0.28
control	Control	control	27.44% $\pm$ 0.73	58.99% $\pm$ 0.65	51.25% $\pm$ 3.30	17.21% $\pm$ 0.93	1.82% $\pm$ 0.27

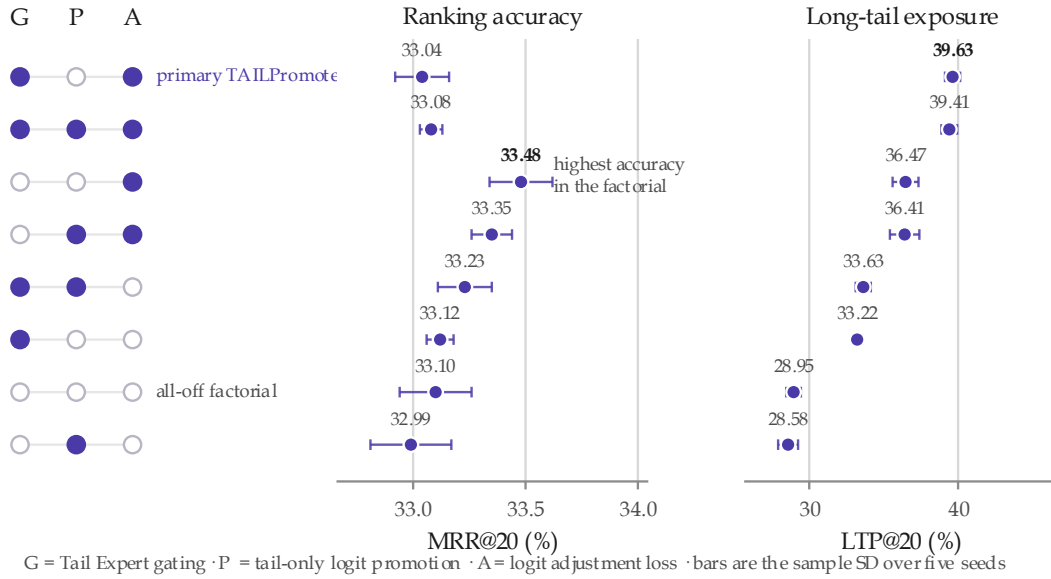


Figure 4. MRR@20 and LTP@20 for all eight factorial configurations. G denotes Tail Expert gating, P denotes tail-only logit promotion, and A denotes logit adjustment; a filled marker indicates that the corresponding switch is enabled. The primary TAILPromote configuration is G=1,P=0,A=1, while G=0,P=0,A=0 denotes the all-off factorial configuration. Error bars show the sample standard deviation over five seeds.

Averaging over the four matched settings of the other two switches, Tail Expert gating changes MRR@20 by -0.11 percentage points and LTP@20 by $+3.87$ points. Tail-only logit promotion changes the same metrics by -0.02 and -0.06 points, respectively, whereas logit adjustment changes them by $+0.13$ and $+6.89$ points. These descriptive marginal effects are summarized in Table 7; their statistical uncertainty is evaluated subsequently using the paired analysis reported in Table 8.

Table 7. Descriptive marginal effects of each factorial switch in percentage points, computed as the mean difference between the on and off settings while averaging over all combinations of the other two switches.

Component	$\Delta\text{MRR@20}$	$\Delta\text{LTP@20}$
Tail Expert gating	-0.11	+3.87
Tail-only logit promotion	-0.02	-0.06
Logit adjustment	+0.13	+6.89

The statistical reliability of the marginal effects was assessed using the paired hierarchical-bootstrap procedure described in Section 3.3. For each switch, the four configuration pairs differing only in that switch were matched, and the resulting marginal effects were evaluated over 10,000 bootstrap replicates with Holm correction across the three switches within each metric. Figure 5 visualizes the effects and their confidence intervals, and Table 8 reports the corresponding numerical results.

Tail Expert gating significantly increases LTP@20 by 3.87 percentage points (95% CI +3.40 to +4.31) and deep-tail exposure by 1.44 points (+1.00 to +1.84). Logit adjustment produces the largest exposure effect, increasing LTP@20 by 6.88 points (+6.49 to +7.28), deep-tail exposure by 4.74 points (+4.34 to +5.15), and Coverage@20 by 2.59 points (+2.22 to +2.94). Tail-only logit promotion shows no statistically significant marginal effect on any reported metric. None of the three switches has a statistically significant marginal effect on MRR@20.

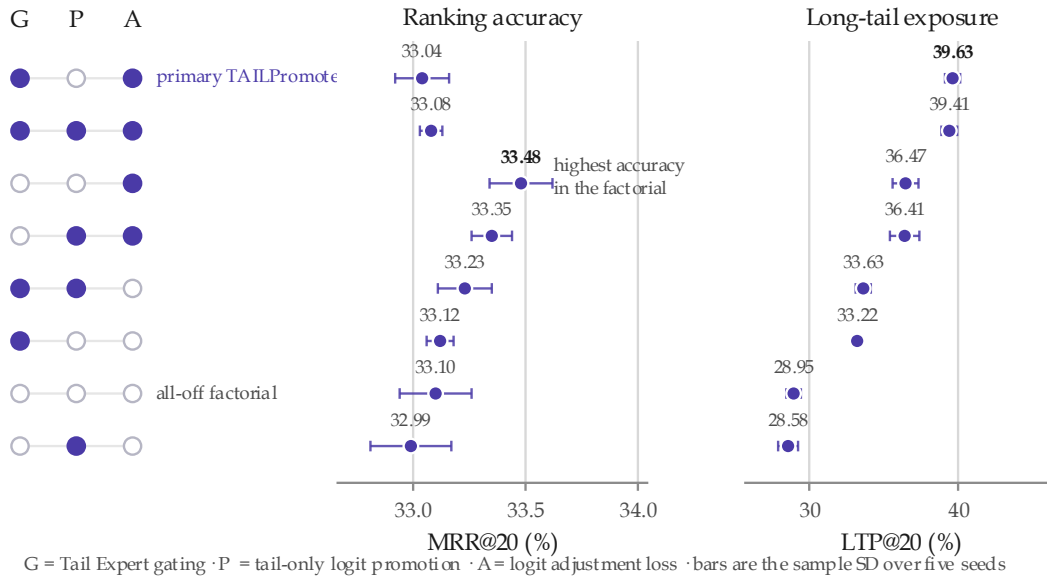


Figure 4. MRR@20 and LTP@20 for all eight factorial configurations. G denotes Tail Expert gating, P denotes tail-only logit promotion, and A denotes logit adjustment; a filled marker indicates that the corresponding switch is enabled. The primary TAILPromote configuration is G=1, P=0, A=1, while G=0, P=0, A=0 denotes the all-off factorial configuration. Error bars show the sample standard deviation over five seeds.

The factorial also shows that component-level contributions do not imply that enabling every available mechanism yields the preferred operating point. The adjustment-only configuration (G=0, P=0, A=1) attains the highest MRR@20 in the factorial at $33.48\% \pm 0.14$, with LTP@20 of $36.47\% \pm 0.87$. The primary TAILPromote configuration (G=1, P=0, A=1) attains MRR@20 of $33.04\% \pm 0.12$ while increasing LTP@20 to $39.63\% \pm 0.51$. Tail-only logit promotion provides no statistically supported independent benefit and is therefore excluded from the primary configuration.

The two effective mechanisms operate at different levels. Relative to TailNet, which adjusts item groups using a session-level preference factor, the Tail Expert provides a candidate-restricted, session-dependent correction controlled by a supervised gate. Relative to NISER+, which mitigates popularity effects through representation normalization, logit adjustment modifies the ranking objective directly using training-derived popularity priors. The factorial results indicate that the objective-level adjustment is the larger contributor to long-tail exposure, while Tail Expert gating provides a smaller but statistically supported complementary effect.

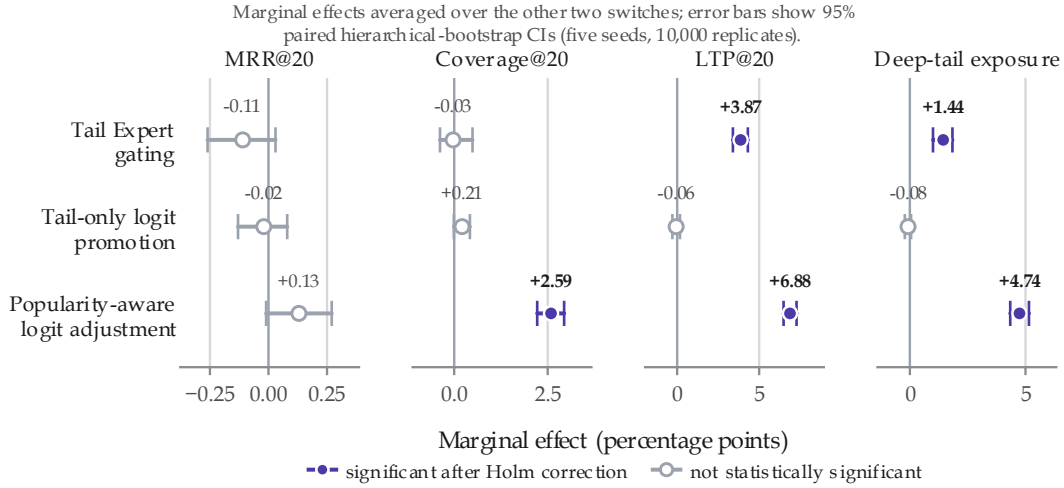


Figure 5. Marginal effects of Tail Expert gating, tail-only logit promotion, and popularity-aware logit adjustment on MRR@20, Coverage@20, LTP@20, and deep-tail exposure. Effects are averaged over the settings of the other two factorial switches, and error bars denote 95% confidence intervals from the paired hierarchical bootstrap over five seeds and 10,000 replicates. Filled markers indicate effects that remain statistically significant after Holm correction; open markers indicate non-significant effects.

Table 8. Marginal effects of the factorial components on the test partition (percentage points), averaged over the settings of the other two switches, with 95% confidence intervals obtained from the paired hierarchical bootstrap over five seeds and 10,000 replicates. -values are Holm-adjusted across the three switches within each metric.

Metric	Component	Effect	95% CI	Holm p	Significant
MRR@20	Tail Expert gating	-0.11	-0.26 to +0.03	0.256	no
MRR@20	Tail-only logit promotion	-0.02	-0.13 to +0.08	0.659	no
MRR@20	Logit adjustment loss	+0.13	-0.01 to +0.27	0.215	no
Coverage@20	Tail Expert gating	-0.03	-0.38 to +0.49	0.809	no
Coverage@20	Tail-only logit promotion	+0.21	-0.01 to +0.42	0.154	no
Coverage@20	Logit adjustment loss	+2.59	+2.22 to +2.94	< 0.001	yes
LTP@20	Tail Expert gating	+3.87	+3.40 to +4.31	< 0.001	yes
LTP@20	Tail-only logit promotion	-0.06	-0.29 to +0.17	0.590	no
LTP@20	Logit adjustment loss	+6.88	+6.49 to +7.28	< 0.001	yes
Deep-tail exposure	Tail Expert gating	+1.44	+1.00 to +1.84	< 0.001	yes
Deep-tail exposure	Tail-only logit promotion	-0.08	-0.22 to +0.05	0.221	no
Deep-tail exposure	Logit adjustment loss	+4.74	+4.34 to +5.15	< 0.001	yes

This pattern is consistent with the role of popularity-aware logit adjustment in changing the relative optimization pressure assigned to items of different popularity. Unlike post-hoc re-ranking approaches [9, 28], the adjustment operates during model training, while the Tail Expert provides a separate session-dependent correction restricted to tail candidates.

4.4. Sensitivity Analysis and Gate Behaviour

We first examine the sensitivity of TAILPromote to the logit-adjustment temperature τ . Table 9 reports validation results across seven values of τ over five seeds, and Figure 6 shows the corresponding accuracy-exposure trade-off.

Table 9. Sensitivity to the logit-adjustment temperature τ on the validation partition over five seeds (%), reported as mean $\pm$ sample standard deviation. A setting is considered admissible when its five-seed mean validation MRR@20 retains at least 95% of the SASRec validation mean and the gate assigns greater mean weight to tail-target than to head-target sessions in every seed.

τ	Val MRR@20	Val Coverage@20	Val LTP@20	Val deep-tail exp.	Rule
0	33.26% $\pm$ 0.15	85.81% $\pm$ 0.44	30.71% $\pm$ 0.27	7.81% $\pm$ 0.13	admissible
0.1	33.36% $\pm$ 0.21	87.31% $\pm$ 0.23	33.54% $\pm$ 0.28	9.57% $\pm$ 0.22	admissible
0.2	33.24% $\pm$ 0.07	88.63% $\pm$ 0.37	36.49% $\pm$ 0.54	11.57% $\pm$ 0.30	admissible
0.4	32.71% $\pm$ 0.12	91.05% $\pm$ 0.23	43.70% $\pm$ 0.98	16.61% $\pm$ 0.66	admissible
0.6	31.85% $\pm$ 0.06	92.52% $\pm$ 0.18	52.21% $\pm$ 0.10	22.30% $\pm$ 0.08	admissible
0.8	30.45% $\pm$ 0.14	93.71% $\pm$ 0.17	58.83% $\pm$ 0.39	26.26% $\pm$ 0.45	admissible
1	28.33% $\pm$ 0.30	94.26% $\pm$ 0.26	63.74% $\pm$ 0.66	29.11% $\pm$ 0.56	below threshold

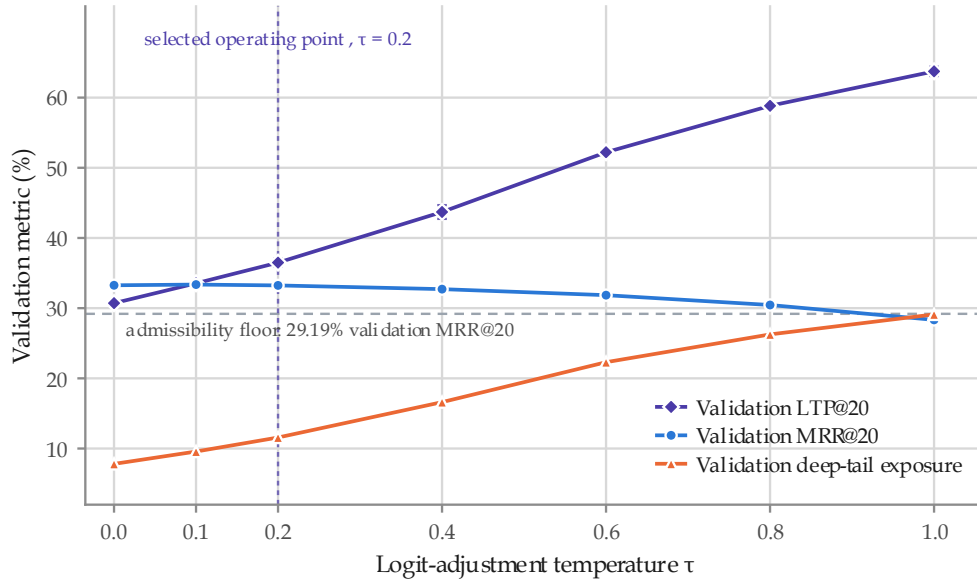


Figure 6. Validation MRR@20, LTP@20, and deep-tail exposure as functions of the logit-adjustment temperature τ across five seeds. Error bars show the sample standard deviation. The horizontal dashed line denotes the predefined MRR@20 admissibility threshold, and the selected operating point is marked at $\tau=0.2$.

The SASRec validation mean is 30.72%, yielding an MRR@20 retention threshold of 29.19%. Values of τ from 0.0 through 0.8 satisfy this threshold, whereas $\tau=1.0$ does not. The gate also assigns greater mean weight to tail-target than to head-target sessions in every seed across the admissible settings. Within this range, increasing τ produces a clear accuracy-exposure trade-off: validation LTP@20 rises from 30.71% $\pm$ 0.27 at $\tau=0.0$ to 58.83% $\pm$ 0.39 at $\tau=0.8$, while validation MRR@20 decreases from 33.26% $\pm$ 0.15 to 30.45% $\pm$ 0.14.

Within the temperature sweep, only the validation-selected setting was subsequently evaluated on the test partition. At the selected operating point, $\tau=0.2$ with $\lambda_{cov}=5$, TAILPromote attains test MRR@20 of 33.04% $\pm$ 0.12, Coverage@20 of 89.93% $\pm$ 0.42, and LTP@20 of 39.63% $\pm$ 0.51, consistent with the primary results in Table 3.

The robustness of the results to the head/tail definition was examined separately by repeating TAILPromote with head catalogue fractions of 5%, 10%, and 20%. This analysis was performed after the primary configuration had been fixed and was not used for model selection. Table 10 reports the resulting test performance, and Figure 7 visualizes the changes across the three definitions.

Table 10. Sensitivity to the head definition on the test partition over five seeds (%), reported as mean $\pm$ sample standard deviation. The long-tail set used by LTP@20 is recomputed at each head fraction, whereas the head, middle-tail, and deep-tail exposure groups remain fixed according to the decile-based definition in Section 3.3.

Head fraction	MRR@20	HitRate@20	Coverage@20	LTP@20	Deep-tail exp.
5%	33.24% $\pm$ 0.15	65.71% $\pm$ 0.30	89.12% $\pm$ 0.17	51.05% $\pm$ 0.67	11.93% $\pm$ 0.28
10%	33.04% $\pm$ 0.12	65.49% $\pm$ 0.35	89.93% $\pm$ 0.42	39.63% $\pm$ 0.51	12.78% $\pm$ 0.38
20%	33.07% $\pm$ 0.08	65.14% $\pm$ 0.09	90.29% $\pm$ 0.33	30.18% $\pm$ 0.08	13.53% $\pm$ 0.18

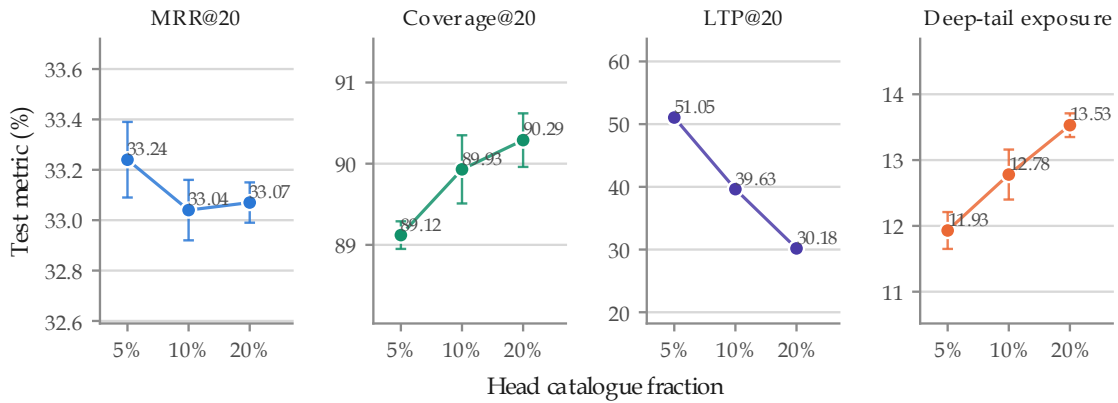


Figure 7. Sensitivity of TAILPromote to head-catalogue fractions of 5%, 10%, and 20% over five seeds. Panels report MRR@20, Coverage@20, LTP@20, and deep-tail exposure as mean $\pm$ sample standard deviation. The long-tail set used by LTP@20 is redefined at each head fraction, whereas the deep-tail group remains fixed according to the decile-based definition in Section 3.3.

MRR@20 remains similar across the three head definitions, ranging from 33.04% $\pm$ 0.12 to 33.24% $\pm$ 0.15. LTP@20 changes substantially, from 51.05% $\pm$ 0.67 at a 5% head fraction to 30.18% $\pm$ 0.08 at 20%, as both the head/tail boundary and the corresponding long-tail denominator are redefined. Coverage@20 varies from 89.12% $\pm$ 0.17 to 90.29% $\pm$ 0.33, while deep-tail exposure ranges from 11.93% $\pm$ 0.28 to 13.53% $\pm$ 0.18. Thus, the same qualitative accuracy-exposure pattern is observed across the three head fractions examined, although the numerical value of LTP@20 necessarily depends on the adopted head/tail definition. We next examine the catalogue-coverage regularization weight λ_{cov} , with τ fixed at its selected value of 0.2. Table 11 reports the validation results across the tested values.

Table 11. Sensitivity to the catalogue-coverage regularization weight λ_{cov} on the validation partition over five seeds (%), reported as mean $\pm$ sample standard deviation, with $\tau=0.2$. Admissibility is assessed using the same validation criterion defined for Table 9.

λ_{cov}	Val MRR@20	Val Coverage@20	Val LTP@20	Val deep-tail exp.	Rule
0	33.12% $\pm$ 0.16	83.38% $\pm$ 0.46	26.30% $\pm$ 0.30	6.35% $\pm$ 0.14	admissible
0.5	33.12% $\pm$ 0.19	84.13% $\pm$ 0.53	27.35% $\pm$ 0.30	6.72% $\pm$ 0.18	admissible
2	33.15% $\pm$ 0.06	86.28% $\pm$ 0.24	31.10% $\pm$ 0.25	8.28% $\pm$ 0.14	admissible
5	33.24% $\pm$ 0.07	88.63% $\pm$ 0.37	36.49% $\pm$ 0.54	11.57% $\pm$ 0.30	admissible
20	32.80% $\pm$ 0.10	92.49% $\pm$ 0.36	48.44% $\pm$ 0.25	20.00% $\pm$ 0.22	admissible
50	32.53% $\pm$ 0.12	94.07% $\pm$ 0.12	54.77% $\pm$ 0.56	24.23% $\pm$ 0.84	admissible
100	32.35% $\pm$ 0.12	94.36% $\pm$ 0.22	56.04% $\pm$ 0.68	24.83% $\pm$ 0.88	admissible

Increasing λ_{cov} progressively broadens catalogue exposure across the tested range. With the coverage penalty disabled, the model attains validation Coverage@20 of $83.38\% \pm 0.46$ and MRR@20 of $33.12\% \pm 0.16$. At the selected weight $\lambda_{cov}=5$, Coverage@20 increases to $88.63\% \pm 0.37$ while MRR@20 is $33.24\% \pm 0.07$. At the largest weight evaluated, $\lambda_{cov}=100$, Coverage@20 reaches $94.36\% \pm 0.22$ and MRR@20 remains at $32.35\% \pm 0.12$. All tested coverage weights remain above the predefined MRR@20 retention threshold.

Figure 8 overlays the two one-dimensional validation sweeps: τ is varied while $\lambda_{cov}=5$, and λ_{cov} is varied while $\tau=0.2$. Both controls move the model toward greater catalogue exposure as their values increase, although the associated changes in ranking accuracy differ across the explored range.

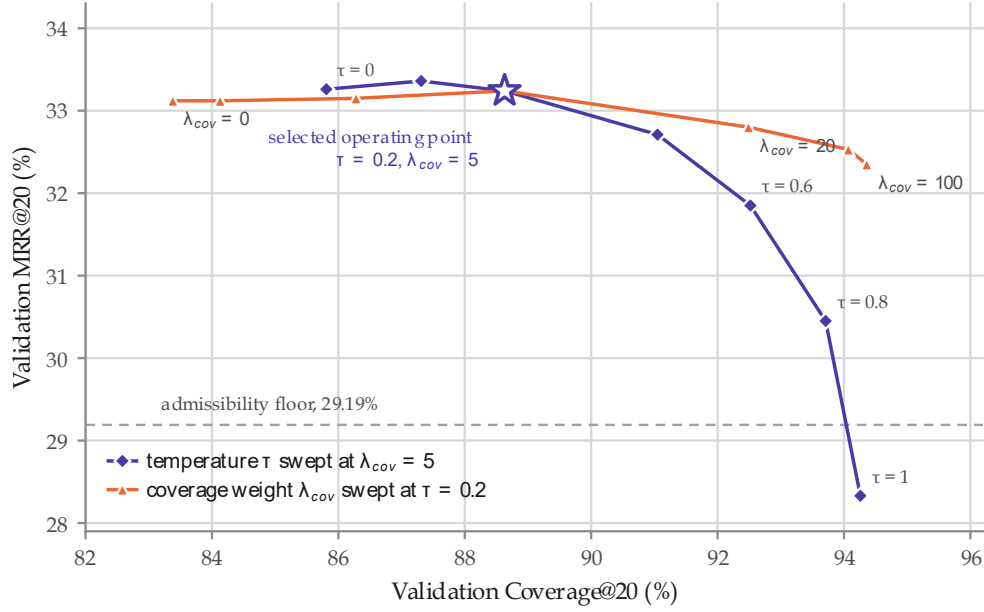


Figure 8. Validation accuracy-coverage trade-off trajectories obtained by varying τ at $\lambda_{cov}=5$ and by varying λ_{cov} at $\tau=0.2$. Values are five-seed means. The horizontal dashed line denotes the predefined MRR@20 admissibility threshold, and the star marks the selected operating point ($\tau=0.2, \lambda_{cov}=5$).

The selected operating point should therefore be interpreted as one validation-selected position within the observed accuracy-exposure trade-off rather than as a globally optimal setting. The configuration used for the primary test results, $\tau=0.2$ and $\lambda_{cov}=5$, satisfies the predefined admissibility criterion while retaining substantially greater exposure than the low-regularization settings. Other admissible settings place greater emphasis on exposure at a larger accuracy cost; these alternatives are reported as validation results only.

We finally examine how the learned gate varies with target-popularity group and with the tail composition of the input session. Because the gate in Equation (4) scales the Tail Expert correction, larger gate values correspond to a greater contribution from the tail-specific pathway.

Grouped by target popularity, the mean Tail Expert gate weight increases from 0.314 for head targets to 0.631 for middle-tail targets and 0.746 for deep-tail targets. Grouped by the tail proportion of the input session, the mean weight increases from 0.282 for sessions containing no tail item to 0.781 for sessions in which more than 75% of the input events are tail items. The group ordering is consistent across all five seeds. Table 12 reports the complete group summaries. Figure 9 shows the distribution of gate weights across the three target-popularity groups, while Figure 10 examines the relationship at finer granularity by target-popularity decile and session tail proportion.

Table 12. Mean Tail Expert gate weight by predefined target-popularity group and input-session tail proportion, averaged over five seeds. The gate is computed from the session representation alone; counts denote test examples per seed.

Grouping variable	Group	Examples	Mean Tail Expert weight
target popularity	Head	11,455	0.314
target popularity	middle-tail	2,719	0.631
target popularity	deep-tail	726	0.746
session tail proportion	0	10,455	0.282
session tail proportion	(0,0.25]	817	0.445
session tail proportion	(0.25,0.50]	1,087	0.560
session tail proportion	(0.50,0.75]	390	0.663
session tail proportion	(0.75,1.00]	2,151	0.781

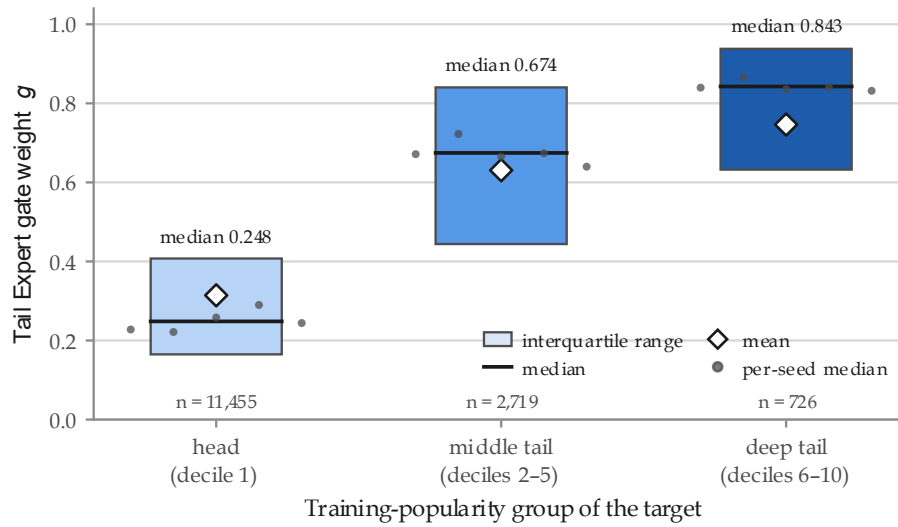


Figure 9. Distributional summaries of the Tail Expert gate weight g by training-defined target-popularity group. Rectangles span the seed-averaged first and third quartiles, horizontal lines denote the seed-averaged medians, diamonds denote the overall mean gate weights, and small points show the five per-seed medians. Counts denote test examples per seed. Whiskers are omitted because the retained summaries contain quartiles rather than full distributional extrema.

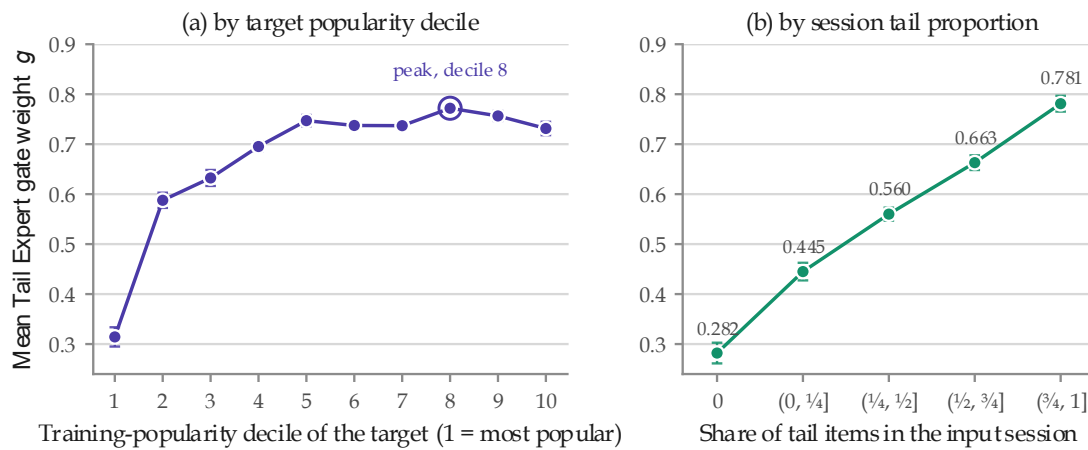


Figure 10. Tail Expert gate weight by target-popularity decile and input-session composition. Panel (a) reports the mean gate weight for each training-popularity decile of the target, with decile 1 denoting the most popular items; panel (b) reports the mean gate weight across bands of the proportion of tail items in the input session. Points are five-seed means and error bars denote sample standard deviations across seeds. The gate is computed solely from the input-session representation and does not observe the target item.

Across seeds, the Spearman correlation between gate weight and negative log training popularity of the target ranges from 0.625 to 0.652, while the correlation between gate weight and session tail proportion ranges from 0.631 to 0.656. The association with target popularity should be interpreted cautiously because the gate does not observe the target and may partly reflect the relationship between the popularity composition of the input session and the popularity of the subsequent target. Moreover, the target-popularity relationship is not strictly monotonic at the decile level: the mean gate weight peaks at decile 8 and decreases slightly thereafter. Accordingly, we report the specialization pattern at the three-group level and do not claim monotonic routing across all ten popularity deciles.

The unconstrained two-expert reference configuration provides an additional comparison with the candidate-restricted design. As reported in Table 4, the unconstrained mixture attains MRR@20 of $27.65\% \pm 0.49$ and LTP@20 of $45.72\% \pm 0.58$, whereas the primary TAILPromote configuration attains $33.04\% \pm 0.12$ and $39.63\% \pm 0.51$, respectively. The corresponding paired differences are +5.39 percentage points in MRR@20 (95% CI +4.92 to +5.78, Holm-adjusted $p < 0.0002$) and -6.09 points in LTP@20 (95% CI -6.97 to -5.32, Holm-adjusted $p < 0.0002$). Thus, the constrained configuration yields substantially higher ranking accuracy, while the unconstrained reference yields greater long-tail exposure. Taken together, the gate analysis indicates a stable association between larger gate weights and less-popular targets or more tail-heavy input sessions, while stopping short of attributing this pattern to latent niche intent.

4.5. Limitations

The findings should be interpreted within several boundaries that define the scope of the present evaluation without altering the main comparative conclusions. First, TAILPromote is not the most accurate model in the study when ranking performance is considered in isolation, since NISER+ achieves an MRR@20 that is 0.90 percentage points higher; however, TAILPromote simultaneously provides markedly greater catalogue coverage, long-tail exposure, and deep-tail exposure while also outperforming SASRec and TailNet in MRR@20. The central result is therefore not that TAILPromote dominates every comparator on every metric, but that it establishes a substantially different operating point in which broad catalogue exposure and long-tail representation are increased while ranking accuracy remains competitive. This distinction is important because the relative value of such an operating point depends on the objective of the recommendation system rather than on a single metric considered independently.

Second, the exposure-oriented metrics used in this study characterize how recommendation opportunities are distributed across the catalogue rather than directly measuring user utility. Coverage@20 quantifies the breadth of the catalogue represented in recommendation lists, while LTP@20 and deep-tail exposure quantify the extent to which recommendation positions are allocated to less-popular items, but none of these measures determines whether an exposed tail item would ultimately be preferred, consumed, or judged useful by a user. This distinction is inherent to offline long-tail evaluation and does not affect the observed redistribution of exposure, but it means that the user-level consequences of the resulting recommendation lists would require complementary online or user-centred evaluation. The gate analysis should be interpreted in the same way: the consistent association between larger gate weights, more tail-heavy input sessions, and less-popular targets establishes a clear behavioural pattern in the learned routing mechanism, while the offline setting does not by itself identify that pattern with an underlying psychological construct such as niche or exploratory intent.

Third, several elements of the experimental protocol intentionally use fixed operational definitions in order to keep comparisons consistent across models. The primary experiments define the head as the most popular 10% of the training catalogue, while the additional analysis at 5%, 10%, and 20% shows that the qualitative accuracy-exposure relationship remains similar across the three definitions examined even though the numerical value of LTP@20 changes with the corresponding tail set. A static popularity boundary nevertheless represents catalogue popularity at the point at which it is estimated and does not model environments in which item popularity evolves continuously over time. The evaluation also

contributes one final-target prediction per eligible session rather than generating a separate prediction for every possible prefix, which makes the reported results specific to the next-item prediction protocol defined in this study while preserving a common and controlled evaluation setting for all compared methods.

Fourth, the empirical evaluation is based on the YooChoose 1/64 benchmark and five random seeds, and the extent to which the same quantitative trade-off is reproduced in datasets with substantially different catalogue sizes, sparsity levels, session-length distributions, or popularity structures remains an empirical question for broader evaluation. The repeated-seed results, paired hierarchical bootstrap analysis, head-definition sensitivity study, and validation sweeps provide evidence that the main patterns are stable within the present experimental setting, but evaluation on additional session-based recommendation datasets would establish how far those patterns extend beyond YooChoose. The search over the auxiliary-loss weights was also deliberately restricted to a small set of paired values with a fixed ratio, and because the selected pair lies at the lower end of that grid, a denser search that includes smaller values could provide a more detailed characterization of the interaction between the auxiliary objectives and the primary ranking loss.

5. Conclusions and Recommendations

This study introduced and evaluated TAILPromote as a tail-aware session-based recommendation framework designed to increase catalogue exposure without relying on a separate post-hoc re-ranking stage. Under the leakage-controlled temporal evaluation protocol and across five random seeds, TAILPromote achieves an LTP@20 of $39.63\% \pm 0.51$, compared with $18.00\% \pm 0.65$ for its SASRec backbone, $24.54\% \pm 0.08$ for NISER+, and $24.25\% \pm 0.18$ for TailNet, with all three paired differences remaining significant after Holm correction, while also attaining the highest Coverage@20 in the study at $89.93\% \pm 0.42$. Its MRR@20 of $33.04\% \pm 0.12$ exceeds SASRec and TailNet and remains below the $33.94\% \pm 0.14$ achieved by NISER+, establishing a clear accuracy–exposure trade-off rather than uniform dominance across objectives. The component factorial further shows that the increase in long-tail exposure is driven primarily by popularity-aware logit adjustment, with Tail Expert gating providing a smaller but statistically supported complementary contribution, whereas tail-only logit promotion shows no independent benefit and is therefore excluded from the primary configuration.

The gate analysis provides additional evidence that the tail-specific pathway is used selectively rather than uniformly across sessions. Larger gate weights are consistently associated with more tail-heavy input sessions and with less-popular subsequent targets, while the gate itself is computed solely from the input-session representation and does not observe the target item. Together with the sensitivity analyses, these results show that TAILPromote provides a controllable mechanism for shifting the balance between ranking accuracy and catalogue exposure through validation-selected settings of the logit-adjustment temperature and coverage regularization weight, while preserving a measurable and interpretable role for the Tail Expert. The resulting contribution is therefore not a claim that long-tail exposure can be increased without cost, but a framework in which that cost can be quantified, compared across configurations, and selected explicitly according to the objective of the recommendation setting.

Unlike multi-stage recommendation pipelines in which an accuracy-oriented model is followed by a separate diversification or re-ranking procedure [9, 28], TAILPromote incorporates its principal long-tail mechanisms directly into model training and scoring, providing an integrated alternative in settings where catalogue exposure is an explicit objective. This is particularly relevant to session-based recommendation, where decisions must often be made from short interaction sequences without persistent user profiles [13, 20], since the gated tail-specific pathway adapts its contribution to the representation of the current session rather than depending on long-term user information. More broadly, the observed redistribution of recommendation exposure toward less-popular items addresses the concentration effects associated with popularity bias [3, 4, 10] and is consistent with multi-objective

recommendation settings in which ranking quality and catalogue-level exposure are considered jointly rather than treated as interchangeable measures of performance [25].

For practitioners, the results support treating TAILPromote as an operating-point model rather than as a fixed accuracy-maximizing replacement for every baseline. Applications that place substantial value on catalogue coverage and long-tail exposure can select τ and λ_{cov} on held-out validation data according to the accuracy–exposure balance required by the application, whereas settings in which ranking accuracy is the dominant objective may favour a model such as NISER+. This distinction is useful because it makes the deployment choice explicit: the model does not conceal the exposure gain behind a single aggregate score, but reports the corresponding ranking cost and allows the balance between the two objectives to be adjusted within the range examined during validation.

Future work should extend the evaluation to additional session-based recommendation datasets, longer temporal windows, and alternative training fractions in order to determine how the observed trade-off changes under different catalogue sizes, sparsity levels, session-length distributions, and popularity structures. Further experiments could also compare the final-target protocol used here with every-prefix training and evaluation, explore denser searches over the auxiliary-loss weights and broader combinations of τ and λ_{cov} , and examine probability-level calibration across the compared models where appropriate. Online or user-centred evaluation would provide the next step toward connecting increased tail exposure with user utility and provider-side outcomes, while recent developments in generative recommendation [29, 30] also motivate investigating whether collaborative and textual representations can be aligned to improve the modelling of under-represented items.

Contribution Statement

All authors contributed equally.

Conflict of Interest Statement

The authors of this article declare that they have no personal or financial conflicts of interest with any institution, organization, or individual.

References

- [1] Pan LW, Pan WK, Wei MY, Yin HZ, Ming Z. A survey on sequential recommendation. *Frontiers of Computer Science* 2026;20(3):2003606. <https://doi.org/10.1007/s11704-025-41329-w>
- [2] Esmeli R, Can AS, Awad A, Bader-El-Den M. Understanding customer loyalty-aware recommender systems in E-commerce: an analytical perspective. *Electronic Commerce Research* 2025;1–27. <https://doi.org/10.1007/s10660-025-09954-6>
- [3] Klimashevskaja A, Jannach D, Elahi M, Trattner C. A survey on popularity bias in recommender systems. *User Modeling and User-Adapted Interaction* 2024;34(5):1777–1834. <https://doi.org/10.1007/s11257-024-09406-0>
- [4] Carnovalini F, Rodà A, Wiggins GA. Popularity Bias in Recommender Systems: The Search for Fairness in the Long Tail. *Information* 2025;16(2):151. <https://doi.org/10.3390/info16020151>
- [5] Park YJ, Tuzhilin A. The long tail of recommender systems and how to leverage it. In: *Proceedings of the 2008 ACM conference on Recommender systems*; 2008. p. 11–18. <https://doi.org/10.1145/1454008.1454012>
- [6] Yu D, Yu T, Wang D, Wang S. Long tail service recommendation based on cross-view and contrastive learning. *Expert Systems with Applications* 2024;238:121957. <https://doi.org/10.1016/j.eswa.2023.121957>
- [7] De Campos LM, Fernández Luna JM, Huete Guadix JF. An explainable content-based approach for recommender systems: a case study in journal recommendation for paper submission. *User Modeling and User-Adapted Interaction* 2024. <https://doi.org/10.1007/s11257-024-09400-6>
- [8] Kumar C, Kumar M. Session-based recommendations with sequential context using attention-driven LSTM. *Computers and Electrical Engineering* 2024;115:109138. <https://doi.org/10.1016/j.compeleceng.2024.109138>

- [9] Peng D, Zhou Y. A long-tail alleviation post-processing framework based on personalized diversity of session recommendation. *Expert Systems with Applications* 2024;249:123769. <https://doi.org/10.1016/j.eswa.2024.123769>
- [10] Liu S, Zheng Y. Long-tail session-based recommendation. In: *Proceedings of the 14th ACM conference on recommender systems*; 2020. p. 509–514. <https://doi.org/10.1145/3383313.3412222>
- [11] Yang H, Choi Y, Kim G, Lee JH. LOAM: improving long-tail session-based recommendation via niche walk augmentation and tail session mixup. In: *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*; 2023. p. 527–536. <https://doi.org/10.1145/3539618.3591718>
- [12] Gupta P, Garg D, Malhotra P, Vig L, Shroff G. NISER: Normalized Item and Session Representations with Graph Neural Networks. In: *1st International Workshop on Graph Representation Learning and its Applications (CIKM '19)*; 2019. <https://doi.org/10.48550/arXiv.1909.04276>
- [13] Hidasi B, Karatzoglou A, Baltrunas L, Tikk D. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*; 2015. <https://doi.org/10.48550/arXiv.1511.06939>
- [14] Chen Q, Jiang F, Guo X, Chen J, Sha K, Wang Y. Combine temporal information in session-based recommendation with graph neural networks. *Expert Systems with Applications* 2024;238:121969. <https://doi.org/10.1016/j.eswa.2023.121969>
- [15] Wang Z, Wei W, Zou D, et al. Exploring global information for session-based recommendation. *Pattern Recognition* 2024;145:109911. <https://doi.org/10.1016/j.patcog.2023.109911>
- [16] Zheng X, Wu R, Han Z, Chen C, Chen L, Han B. Heterogeneous information crossing on graphs for session-based recommender systems. *ACM Transactions on the Web* 2024;18(2):1–24. <https://doi.org/10.1145/3572407>
- [17] Sheng Z, Zhang T, Zhang Y, Gao S. Enhanced graph neural network for session-based recommendation. *Expert Systems with Applications* 2023;213:118887. <https://doi.org/10.1016/j.eswa.2022.118887>
- [18] Shafiloo R, Kaedi M, Pourmiri A. Considering user dynamic preferences for mitigating negative effects of long tail in recommender systems. *Information Sciences* 2024;669:120558. <https://doi.org/10.1016/j.ins.2024.120558>
- [19] Zhao Z, Zhou K, Wang X, Zhao WX, Pan F, Cao Z, Wen JR. Alleviating the long-tail problem in conversational recommender systems. In: *Proceedings of the 17th ACM conference on recommender systems*; 2023. p. 374–385. <https://doi.org/10.1145/3604915.3608812>
- [20] Jangid M, Kumar R. Deep learning approaches to address cold start and long tail challenges in recommendation systems: a systematic review. *Multimedia Tools and Applications* 2025;84(5):2293–2325. <https://doi.org/10.1007/s11042-024-20262-3>
- [21] Didi T, Guy I, Livne A, Dagan A, Rokach L, Shapira B. Promoting Tail Item Recommendations in E-Commerce. In: *Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization*; 2023. p. 194–203. <https://doi.org/10.1145/3565472.3592968>
- [22] Lee G, Kim K, Shin K. Post-Training Embedding Enhancement for Long-Tail Recommendation. In: *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*; 2024. p. 3857–3861. <https://doi.org/10.1145/3627673.3679978>
- [23] Zhang Y, Cheng DZ, Yao T, Yi X, Hong L, Chi EH. A model of two tales: Dual transfer learning framework for improved long-tail item recommendation. In: *Proceedings of the web conference 2021*; 2021. p. 2220–2231. <https://doi.org/10.1145/3442381.3450086>
- [24] Gong Z, Wu X, Chen L, Zheng Z, Wang S, Xu A, et al. Full index deep retrieval: End-to-end user and item structures for cold-start and long-tail item recommendation. In: *Proceedings of the 17th ACM Conference on Recommender Systems*; 2023. p. 47–57. <https://doi.org/10.1145/3604915.3608773>
- [25] Li W, Lyu F, Shang F, Wan L, Feng W. Long-tailed learning as multi-objective optimization. In: *Proceedings of the AAAI Conference on Artificial Intelligence*; 2024. p. 3190–3198. <https://doi.org/10.1609/aaai.v38i4.28103>
- [26] Kang WC, McAuley J. Self-attentive sequential recommendation. In: *Proceedings of ICDM 2018*; 2018. <https://doi.org/10.1109/ICDM.2018.00035>
- [27] 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*; 2015. p. 357–358. <https://doi.org/10.1145/2792838.2798723>
- [28] Wu J, Chang CC, Yu T, He Z, Wang J, Hou Y, McAuley J. Coral: collaborative retrieval-augmented large language models improve long-tail recommendation. In: *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*; 2024. p. 3391–3401. <https://doi.org/10.1145/3637528.3671901>
- [29] Xin H, Sun Y, Wang C, Xiong H. Llmcdsr: Enhancing cross-domain sequential recommendation with large language models. *ACM Transactions on Information Systems* 2025. <https://doi.org/10.1145/3715099>

- [30] Lu Q, Lyu H, Zheng J, Wang Y, Zhang L, Zhou C. Research on E-Commerce Long-Tail Product Recommendation Mechanism Based on Large-Scale Language Models. In: Proceedings of the 9th International Conference on Electronic Information Technology and Computer Engineering; 2025. p. 997–1002.
<https://doi.org/10.1145/3766671.3766843>