Keeping the Index Open: The Recommendation-Side Cost of Shared Search and Recommendation

Theodore Rogers
theobrog@amazon.com
Amazon Web Services
United States

Joe Standerfer
joesta@amazon.com
Amazon Web Services
United States

Dmitrii Timoshenko
timodm@amazon.com
Amazon Web Services
United States

Haoxue Li
lihaoxue@amazon.com
Amazon Web Services
United States

Zuhaib Akhtar
zuhaibak@amazon.com
Amazon Web Services
United States

Soyoung Yang
ysoyoung@amazon.com
Amazon Web Services
United States

Abstract

A shared search-and-recommendation index has to score new items from their features alone, because search has no exploration slot in which to warm them up. On a public log carrying both surfaces over one catalog, 38.6% of query-search impressions in a held-out week show an item never previously shown or visited. On a narrower slice — engagements where the user, not the catalog, had no prior contact with the item — the feature-based tower serves that demand at no measurable penalty against 99 sampled negatives (0.9595 Recall@20 against 0.9510 warm), a parity a lexical baseline shares and a full-catalog check leaves statistically undecided. Dual-encoder (two-tower) retrieval keeps the index *open* in exactly this sense, where an ID-softmax recommender admits a new item only by retraining. We price that openness on the recommendation side — the half that admits controlled offline evaluation — against six sequential baselines, each retrained and tuned through five documented rounds on corrected targets (a float32 timestamp bug in a standard library silently reorders the leave-one-out target for 19.7% of users). Warm accuracy costs 5.2% Recall@20 relative (11.4% NDCG@20) against the strongest retrained baseline in each metric on MovieLens-1M (3.4K items); on MIND (9.4K items) the gap narrows to 0.8–3.6% of the five strongest baselines, though sixth of seven rows. Cold coverage is the payoff, and the shared index's tightest constraint: under a strict cold-start protocol with zero cold-item leakage the content tower reaches 0.172 ± 0.006 Recall@20, 1.4× the strongest of the three dedicated methods we could retrain (0.124 ± 0.007) and 3× a training-free floor, with no cold-start-dedicated training stage. The training objective hides a quality–scale trade-off. Swapping a deployed sampled InfoNCE recipe for exact full-softmax cross-entropy (loss, scoring, sampling and schedule move together) lifts test Recall@20 by 54% on MIND-small and 6.9% on MovieLens-1M, one matched pair per benchmark. But the exact objective recomputes the item tower catalog-wide every step and exhausts accelerator memory at a 240K-item catalog in our implementation, the scale that motivates the architecture. Approximate nearest-neighbor search contributes nothing to the gap and retrieval-stage serving cost does not regress against the ID-softmax class; a history-window sweep attributes half the post-recipe remainder. Exact-quality training at catalog scale is the open problem this leaves.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
USRW@RecSys '26, Minneapolis, MN, USA
© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-XXXX-X/2026/10
https://doi.org/XXXXXXX.XXXXXXX

CCS Concepts

• **Information systems** → **Recommender systems**; *Information retrieval*; • **Computing methodologies** → Neural networks.

Keywords

sequential recommendation, two-tower model, bi-encoder retrieval, cold-start, training objectives, evaluation, unified search and recommendation

ACM Reference Format:
Theodore Rogers, Joe Standerfer, Dmitrii Timoshenko, Haoxue Li, Zuhaib Akhtar, and Soyoung Yang. 2026. Keeping the Index Open: The Recommendation-Side Cost of Shared Search and Recommendation. In *Proceedings of USRW: Workshop on Unified Search and Recommendation at ACM RecSys 2026 (USRW@RecSys '26).* ACM, New York, NY, USA, 12 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Production platforms increasingly serve search and recommendation from one shared item index: one set of pre-computed item embeddings answers both query-initiated retrieval (query tower) [12] and history-initiated retrieval (user tower) [1, 51]. What this shared dual-encoder architecture costs on the *recommendation* side has not been quantified under controlled conditions. We supply that account with retrained baselines under one evaluation convention. The headline finding is a quality–scale trade-off in the training objective: exact full-softmax cross-entropy is the largest accuracy lever we can isolate by paired experiment, yet in our implementation it stops fitting in accelerator memory at the 10^5-plus-item catalogs a shared index serves (§4.3). We measure the recommendation-side cost of an architecture search-side systems already deploy; search-side training and evaluation of the trained towers is a natural next step. They bind on the search side too, cold coverage most of all. A recommender can *bootstrap* a cold item by exploring (placing it in slates to harvest the interactions that make it warm) and omit it silently meanwhile; search results are held to per-query relevance,

so injecting under-served items to collect feedback is not available there. A lexical index covers a just-ingested item from the first query; what a closed-vocabulary retriever cannot do is contribute ranking signal for it, so the dense channel of a hybrid system is dark on new inventory precisely when the lexical channel is least discriminative. Cold items must therefore be scored on content from the first request, and a shared index inherits the stricter requirement (§4.4). The same per-step catalog-linear compute applies to any pipeline that recomputes the shared item tower catalog-wide during training (§4.3), and the serving curve to every consumer of the index (§4.5).

Transformer-based sequential recommenders such as SASRec and BERT4Rec achieve strong accuracy in next-item prediction by applying self-attention over user behavior sequences [14, 37]. These models classify over the item vocabulary (one trained output vector per item ID), a class we call *ID-softmax* models. The production consequence is a *closed index*: one parameter vector per training-time item ID, so unseen items cannot be scored even with rich features at inference [31, 40] and catalog growth is served only by retraining. The catalog-sized output scan is a second cost, linear in N and dominant at million-item catalogs [28], but not an asymmetry: a dual encoder scans its index with the same matmul, so we report serving as a cost ledger (§4.5) and study the closed index as the structural difference.

RetrievalFormer reframes next-item recommendation as retrieval. A transformer user tower encodes the interaction history into a user embedding, a feature-based item tower encodes items from their attributes, and both towers are trained jointly so that the relevance score is a dot product in a shared space,

$$s(u, i) = f_u(\text{history}_u)^\top f_i(\text{features}_i).$$

At serving time, top-K recommendations are produced by (approximate) nearest-neighbor search over pre-computed item embeddings [13, 23], and new items are scored zero-shot from their features. Neither property is free. We quantify the accuracy the architecture concedes to sequential full-softmax recommenders and why (a training-objective constraint that binds at unified-catalog scale, the 10^5-plus-item inventories a shared search-and-recommendation index serves), and the capabilities it gains: cold items served through the same feature-encoded tower that search-side matching requires.

Our contributions:

(1) **The accuracy cost and its scaling wall.** Paired experiments isolate the sampled-to-full-softmax recipe swap as the largest single lever, and characterize the memory wall that puts the exact objective out of reach at unified-catalog scale in our implementation. Against six baselines retrained on a corrected-target benchmark and tuned through five documented rounds each, RetrievalFormer reaches 94.8% of the strongest on MovieLens-1M (0.3739 ± 0.0013, 3 seeds); a history-window sweep attributes half the remaining gap (§4.3).

(2) **A strict cold-start stress test: the shared index's binding constraint.** An ID-softmax model's closed vocabulary admits new items only by retraining, so on a shared index the retraining cadence is the item-availability latency for *both* surfaces. Under an item cold-start split with zero cold-item leakage, we retrain and tune three *dedicated cold-start baselines* under the same protocol (DropoutNet, Heater, ALDI). The feature-based tower exceeds the strongest of them on cold Recall@20 by 1.4× and the rest by 1.6–1.8×, and sits 3× above a training-free content-similarity floor; a fourth dedicated method is cited as prior art only, its implementations having not cleared licensing review (§5). A simple frozen score fusion with tuned matrix factorization (3 seeds) sits above all three on both the cold and warm axes (§4.4).

(3) **Serving cost and protocol diagnostics.** We report exact-scan and IVF-PQ latency with the ANN-vs-exact top-K recall it costs, plus the same measurement on the retrained ID-softmax baseline; the two classes' retrieval-stage cost is symmetric (§4.5). We also report seen-item echo rates for every model — how many already-seen items appear in the top 20 — a diagnostic rarely surfaced in model comparisons. Finally, we audit the evaluation targets: float32 timestamp quantization in a standard benchmarking library silently reorders the leave-one-out target (each user's chronologically last interaction) for 19.7% of MovieLens-1M users, so cross-model comparisons on this benchmark are not directly comparable across pipelines, and retraining on the corrected benchmark lowers every baseline score by 0.016–0.039 (§4.1, §4.2).

(4) **The measurement vehicle.** RetrievalFormer: a dual-encoder sequential recommender with an attention-based heterogeneous feature encoder (AttentionFusion) and embedding tables shared across towers, and zero-shot scoring of unseen items, used here for ablation, not as a novelty claim (§3).

2 Related Work

Sequential recommendation and transformers. Sequential recommenders model user-item interaction sequences to predict the next item. Early approaches used Markov chains or RNNs [9, 19, 38, 45]; recent methods are dominated by self-attention. SASRec [14] uses unidirectional transformer layers; BERT4Rec [37] uses bidirectional attention with a Cloze objective. These models predict via a softmax over the item vocabulary; the head serves as inner-product scoring, but its vocabulary is closed: items enter it only by training [36].

Two-stage retrieval. Industry-scale systems retrieve candidates with lightweight models (matrix factorization, two-tower networks) before precise ranking [1, 4, 7, 12, 51]. PinnerFormer [26] is the closest deployed relative of our setup, a transformer user tower trained with sampled softmax over pretrained item embeddings for retrieval at Pinterest, reported through deployment metrics, not controlled offline comparison. Sampled and corrected softmax objectives for large-corpus retrieval [17, 49] are central in this setting. For single-tower ID-softmax models loss choice is decisive: switching SASRec from sampled BCE to full cross-entropy flips reported model rankings on standard benchmarks [16], negative sampling induces correctable overconfidence [27], and sampled softmax with enough corrected negatives approaches the full objective [47]. A single dot-product similarity head has a separate,

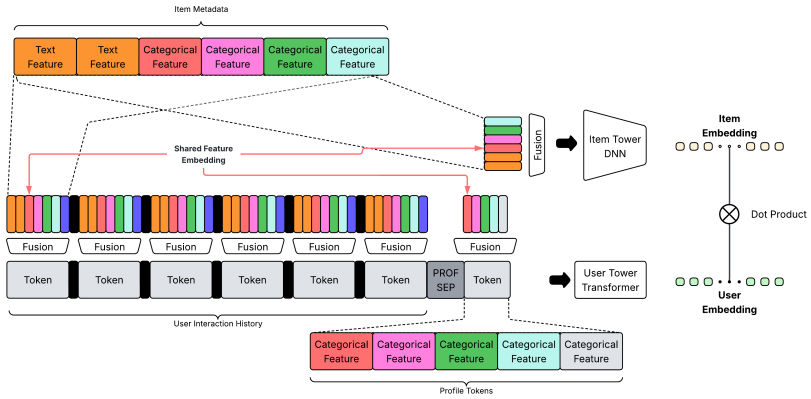


Figure 1: RetrievalFormer data flow. Shared embedding tables (red) embed each categorical value identically wherever it appears: in item metadata, inside a history event, or as a user attribute. Item side: AttentionFusion pools the item's embedded features and the item MLP produces a pre-computable, indexable item embedding (f_i). User side: each interaction is fused into one *enriched token*; the sequence [history, [SEP], profile token] feeds the user-tower transformer (prefix-LM: bidirectional within the appended special-token block, causal over history), read out at a final readout token (not drawn) into f_u. Relevance is the dot product. The item-ID input pathway with its tied output head (§4.1) is not drawn.

documented expressiveness ceiling [36, 53]. We measure the objective effect in the dual-encoder setting, where the exact objective also hits a catalog-bound feasibility wall (§4.3).

Unified search and recommendation. Search and recommendation are also increasingly modeled jointly: a single model can learn retrieval and recommendation from the same user–item interactions [52], and search-behavior signals lift sequential recommendation when the two behavior streams are disentangled or their transitions modeled explicitly [32, 33]. Other work unifies search and recommendation *generatively*: semantic-ID models autoregressively decode item identifiers [30], replacing catalog-wide scoring with constrained decoding over compact codeword vocabularies, restructuring exactly the catalog-linear term behind the memory wall of §4.3, and hybrid designs interpolate between dense and generative retrieval [50]. Our study prices the dual-encoder architecture under controlled retrained-baseline protocols.

Attribute-enriched and cold-start recommendation. Side information improves recommendation under sparsity [2, 26, 56]. AttrFormer [21] models item attributes inside an ID-softmax transformer; like all ID-softmax models, it cannot score unseen item IDs. UniSRec [10] learns transferable sequential representations from item text, sharing the premise that content-based encoders unlock unseen items; we price that premise's dual-encoder serving form under controlled protocols. Dedicated item cold-start methods score cold items from content: DropoutNet [40] trains with representation dropout, Heater [57] mixes randomized training with expert transformations, and contrastive and distillation variants followed [11, 43]. Comparing a feature-based model only against ID-softmax models (which trivially score zero on cold items) would overstate its contribution, so we retrain DropoutNet, Heater, and ALDI under our strict split and tune each through documented rounds (§4.4).

Evaluation rigor. Reported gains often shrink against properly trained baselines [5], so we retrain and tune every baseline under one protocol and report a seen-item echo diagnostic quantifying the evaluation-time masking choice that materially moves all models' scores (§4.1).

Approximate nearest neighbor search. IVF, HNSW, and PQ enable millisecond vector search over huge collections [13, 23]. We propose no new index; we measure what an aggressive IVF-PQ configuration costs in fidelity against exact scoring of the same embeddings (§4.5).

3 Methodology

RetrievalFormer is a dual-encoder architecture with asymmetric towers (Figure 1). The item tower $f_i(\cdot)$ encodes each item's heterogeneous features into a dense embedding that can be pre-computed and indexed.

The user tower $f_u(\cdot)$, a transformer, processes the interaction sequence into a query embedding. Recommendations maximize the dot product between the two. Notation: we write d for the shared output embedding dimension in which user and item vectors meet; the transformer width d_{model} and per-feature embedding sizes d_f are separate hyperparameters (configurations in §4.1).

3.1 Attention Fusion for Heterogeneous Features

Recommendation data mixes single-valued categorical features (e.g., category), multi-valued features (e.g., tags), and text-derived token features. Given features $\mathcal{F} = \{f_1, \ldots, f_M\}$ describing an entity, we embed each feature with shared lookup tables and project to a common dimension:

$$\mathbf{H} = [\mathbf{W}_1\mathbf{E}_{f_1}(f_1); \ldots; \mathbf{W}_M\mathbf{E}_{f_M}(f_M)] \in \mathbb{R}^{M \times d_{\text{model}}}. \quad (1)$$

AttentionFusion is a Set-Transformer-style encoder block [18, 39]: multi-head self-attention over the feature set with residual connections and layer normalization, followed by pooling,

$$\mathbf{Z} = \mathrm{LayerNorm}(\mathbf{H} + \mathrm{MultiHeadAttn}(\mathbf{H}, \mathbf{H}, \mathbf{H})), \tag{2}$$

$$\mathbf{U} = \mathrm{LayerNorm}(\mathbf{Z} + \mathrm{FFN}(\mathbf{Z})), \qquad \mathbf{z} = \mathrm{MeanPool}(\mathbf{U}). \tag{3}$$

The block is permutation-invariant, handles variable-length inputs, and learns how features interact [34]; the same block fuses item metadata, interaction context, and the user profile.

3.2 Shared Embedding Design

Embedding tables are shared across towers: when a categorical value (e.g., a category) appears as an item attribute, inside the user's history, or as a user attribute, it uses the same embedding vector. This reduces embedding parameters (roughly 3× in our implementation), keeps feature semantics consistent between towers, and helps cold-start generalization: a new item's features are interpreted by embeddings already trained from every context in which those features occurred. Item representations trained under the recommendation objective remain indexable for query-initiated retrieval from the same unified catalog; whether they reach competitive ad-hoc relevance without a search-trained query tower is untested here (§5).

3.3 Item and User Towers

Item tower. $\mathbf{y}_i = \mathrm{AttentionFusion}(\mathcal{F}_i) \in \mathbb{R}^d$ depends only on item-side features, so all item embeddings are pre-computed offline and indexed; new items receive embeddings from their features without retraining.

User tower. Each historical interaction is encoded as one input token we call an *enriched token*: the item's fused features concatenated with embeddings of that interaction's own context (rating and temporal fields), projected to d_{model}. The transformer processes the sequence

$$\mathrm{S} = [\mathbf{z}_1, \ldots, \mathbf{z}_T, [\mathrm{SEP}], \mathbf{p}_u, \mathrm{tok}_{\text{out}}], \tag{4}$$

where $\mathbf{p}_u$ fuses static user attributes and $\mathrm{tok}_{\text{out}}$ is the readout position whose final representation, projected to $\mathbb{R}^d$, is the user embedding. Our best configuration uses prefix-LM attention with a last-token readout and additionally feeds item-ID embeddings as an input feature with a tied output head (§4.1); the ID pathway helps warm accuracy and is absent for cold items by construction.

3.4 Training Objectives

We train the shared embedding space with one of two *recipes*: a bundle of four co-moving choices — loss, scoring function, negative sampling, and learning-rate schedule — that production pipelines change together. §4.3 measures their difference.

Sampled InfoNCE with mixed negatives. The standard dual-encoder objective [25, 51]: for a batch of B user-item pairs,

$$\mathcal{L}_{\text{InfoNCE}} = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp(\mathbf{x}_i^\top \mathbf{y}_i/\tau)}{\sum_{j\in\mathcal{N}_i} \exp(\mathbf{x}_i^\top \mathbf{y}_j/\tau)}, \tag{5}$$

where $\mathcal{N}_i$ contains the in-batch items plus k uniformly sampled catalog items (mixed negative sampling [49]), and τ is a temperature. InfoNCE implicitly balances alignment of positive pairs against uniformity of the embedding distribution [42], which counteracts representation collapse and suits ANN indexing. Its per-step cost is independent of catalog size.

Full-softmax cross-entropy. When the catalog fits in accelerator memory, we instead score *all* items each step with an unnormalized dot product plus a learned per-item bias and train with cross-entropy over the full catalog. This optimizes the full ranking directly. Its per-step memory scales with catalog size, because the entire item tower output (with gradients) is materialized every step; §4.3 shows this becomes infeasible surprisingly early.

4 Experiments

We ask: **RQ1** How close does RetrievalFormer come to retrained ID-softmax transformers on warm next-item accuracy? **RQ2** Which measured factors contribute to the remaining gap, and how much remains unattributed? **RQ3** What does the feature-based item tower deliver under a strict item cold-start protocol, against dedicated cold-start baselines? **RQ4** Does approximate search contribute to the accuracy gap, what does it cost in fidelity, and how does serving cost compare against the ID-softmax class?

4.1 Setup and Protocol Diagnostics

Data. MovieLens-1M with 5-core filtering: 6,040 users, 3,416 items, 999,611 interactions; leave-one-out splits (last item test, second-to-last validation). For the paired objective experiment in RQ2 we additionally use the MIND news dataset [46] (small edition, ~9K interacted items after preprocessing), whose item text (titles, abstracts) exercises the feature-based tower; the paired comparison holds data, features, architecture, and training budget identical within that benchmark.

Metrics and convention. Full-catalog ranking, Recall@20 and NDCG@20, no negative sampling at evaluation on the three recommendation benchmarks (the search-side read of §4.4 is the one exception, labelled in place). Echo@20 (the average count of already-seen items in a model's top 20) is reported alongside as a protocol diagnostic. We do *not* mask previously seen items from the candidate set (the common convention for these baselines); Table 1 reports it per model, and §4.2 quantifies what the convention choice means for scores. All models are compared under the identical convention; we call this the *exact convention* (exhaustive full-catalog scoring, seen items retained as candidates), and RetrievalFormer's values are exact-convention rescores of its training checkpoints (Table 1 note). Disabling a residual candidate mask in its harness lowers the affected RetrievalFormer scores to the values reported here; it alters no model weights and no baseline score.

Target-definition audit. Leave-one-out's "last item" is not framework-invariant: RecBole 1.2.1 downcasts timestamps to float32 on load, which ties same-sitting interactions at MovieLens epochs (~64–128 s resolution) and resolves the ties by stable sort over file order, so its *realized* LOO target differs from the chronologically last item for 1,190/6,040 (19.7%) of ML-1M users. We verified the mechanism exactly: a float32-quantized stable sort reproduces RecBole's realized targets for every user. Our pipeline's targets are certified chronologically-last (under exact integer timestamps; agreement asserted by build gates) for 100% of users on every benchmark we print, on both test and validation splits. The baselines of Table 1 are

retrained end-to-end on a corrected benchmark whose build gate asserts realized target = chronological last for 6,040/6,040 users on both splits. Every row of the table therefore shares one target definition. §4.2 quantifies what the correction changes: every baseline score falls (−0.016 to −0.039), and intermediate checkpoint re-scores bound the components.

Baselines. We retrain every baseline ourselves in RecBole [55] with full-softmax cross-entropy, then tune each through five documented rounds: a published-config retrain (R0) followed by one configuration lever per round (learning rate, dropout, capacity, masking ratio, batch size), selected on validation NDCG@10 with ties retained by the incumbent, each round recorded in a per-model ledger. Champions on the corrected benchmark: the ID-only sequential models GRU4Rec [9] (lr 2×10^{-3}), SASRec [14] (lr 5×10^{-4}, dropout 0.3), and BERT4Rec [37] (lr 5×10^{-4}, mask 0.2), and three side-information sequential models consuming item genres: SASRecF (RecBole's feature-fused SASRec; lr 5×10^{-4}, dropout 0.3), FDSA [54] (lr 5×10^{-4}), and DIF-SR [48] at its published 1×10^{-4} (the swept base rate collapses it on every benchmark we tested; Table 1 note). Baseline evaluation is validated by an independent NDCG@10 re-scoring self-check (absolute deltas $\leq 3.4\times10^{-5}$ against the training harness's aggregate). Baseline table rows are 3-seed means (FDSA: single run; Table 1 note); RetrievalFormer's table row is likewise a 3-seed mean; unless stated otherwise, ± denotes sample standard deviation over the stated seeds. Input parity: no model in Table 1 uses title text; the ID-only baselines consume item IDs alone, SASRecF, FDSA, and DIF-SR add item genres, and RetrievalFormer adds item attributes and user metadata. The information asymmetry between RetrievalFormer and the ID-only baselines favors our model. Title ablations are architecture-dependent, over a complete matrix of three architectures by three benchmarks (nine matched single-seed pairs, −0.005 to +0.010): titles lift FDSA, the one baseline with a dedicated feature-sequence attention branch, by +0.0115 (0.3767 → 0.3882, single seed, replicating in direction on a second MovieLens benchmark), and are inert-to-harmful for SASRecF (0.3390 → 0.3372 on MIND) and DIF-SR (within ±0.002) on every bed. Every title delta is an order of magnitude under the same models' tuning gains (+0.03 to +0.11 on MIND-large), so token-sequence titles are a second-order feature for these architectures, where the dual encoder reads the same text through a pretrained encoder. Table 1 keeps every baseline at its best no-title configuration to preserve the stated input parity; text content pays off most unambiguously in cold-start (§4.4). AttrFormer [21], an attribute-aware ID-softmax transformer whose implementation is not publicly available, reports 0.413 Recall@20 on MovieLens-1M; we cite it as reported.

RetrievalFormer configuration. The submitted configuration, selected from sweeps over schedule, dropout, learning rate, batch size, capacity, depth, and history cap, uses prefix-LM attention, last-token readout, and item-ID input with a tied output head. Architecture: d_{model}=256, FFN 512; AdamW (weight decay 5×10^{-4}, none on embeddings) with per-tower learning rates (user 3×10^{-4}, item 1×10^{-3}) on a cosine schedule, best-validation checkpoint, bf16. The single-rate constant-LR variant scores 0.3668 ± 0.004 (5 seeds) and carries the ablations of §4.3. Trained with the full-softmax objective (§3.4) and feature-noising regularization[1]; the sampled-objective counterpart appears in RQ2.

Table 1: MovieLens-1M, full-catalog unmasked ranking, own runs only, all rows sharing one target definition on the corrected chronological-last benchmark (§4.1). Echo@20 = average count of already-seen items in the top 20, a protocol diagnostic (§4.2); RetrievalFormer's is scored at seen-window parity against the user's 50 most recent training items, the window the baselines consume. Mean±std over 3 seeds, each baseline at its best of a published-config retrain plus five validation-selected tuning rounds; †FDSA is a single run (its 3-seed set exists only for the genre+title variant, 0.3838±0.0043**, which the no-title input-parity rule excludes). RetrievalFormer's seeds are truly-unmasked exact-convention GPU rescores of its shipped checkpoints (per-seed Recall@20** 0.3748/0.3745/0.3724**). DIF-SR prints its published-learning-rate configuration, which maximizes test Recall@20; its validation-selected round trades Recall for NDCG (**0.3881 ± 0.0024/0.1910 ± 0.0014**), and ratio statements in the text use the conservative** 0.3944 **(against the validation-selected round the warm cost would be** 3.7%**, not** 5.2%**). Best per column bold, runner-up underlined.**

Model	Recall@20	NDCG@20	Echo@20
GRU4Rec	0.3687±0.0039	0.1799	3.11
SASRec	<u>0.3893±0.0024</u>	0.1876	3.87
BERT4Rec	0.3453±0.0024	0.1542	4.22
SASRecF (genres)	0.3851±0.0015	**0.1911**	3.33
FDSA (genres)†	0.3767	0.1742	4.12
DIF-SR (genres)	**0.3944±0.0019**	<u>0.1883</u>	3.99
RetrievalFormer (history 300)	0.3739±0.0013	0.1693	4.42

4.2 RQ1: Warm Accuracy Against Retrained Baselines

RetrievalFormer recovers most of the strongest baseline's warm accuracy. It reaches Recall@20 0.3739 ± 0.0013 (3 seeds) against retrained-and-tuned baselines (Table 1): 94.8% of the strongest (DIF-SR, 0.3944 ± 0.0019; the NDCG@20 cost is larger, 0.1693 against SASRecF's 0.1911, 11.4%), within 0.003 of FDSA (0.3767) and 0.011 of SASRecF (0.3851), above GRU4Rec (0.3687) and BERT4Rec (0.3453), and 0.015 under tuned SASRec (0.3893), the model tuning moved most (+0.04 over its published-config retrain). All baselines are retrained and tuned under one documented protocol rather than copied from prior reports: retraining with the full-softmax objective moves them substantially, consistent with the loss-choice findings of [16], and the five tuning rounds move four of the six again (+0.009–0.040). The remaining gap accompanies an architecture whose item side is a feature function, open to unseen items; RQ2 separates its measured recipe and history-context legs from a small

[1] Feature noising: during training, 5% of observed categorical feature values are replaced with the reserved unknown token before embedding lookup (element-wise for multi-valued bags), so the unknown-token embedding trains regularly and the partially-unknown cold-item case is in-distribution at inference [3, 35].

unattributed remainder, and RQ3/RQ4 measure what the architecture buys.

The correction, quantified. Retraining end-to-end on the corrected benchmark is the definitive comparison, and it moves the standings: every baseline falls by 0.016–0.039 (published-configuration retrains, R0: GRU4Rec 0.3581, SASRec 0.3498, BERT4Rec 0.3296, SASRecF 0.3757, FDSA 0.3767, DIF-SR 0.3924), the ordering among the baselines shifts (FDSA overtakes SASRecF and GRU4Rec), and the model-vs-baseline ordering changes: our model moves from exceeding one baseline to exceeding three of the six. Those counts are against the published-configuration retrains (R0); the tuning rounds behind Table 1 then lift four of the six corrected baselines back up by +0.009–0.040, and the standings of record are Table 1's. Two diagnostics confirm the direction without being clean reference values, since the affected targets were *training positives* for the original checkpoints: on the original realized targets the same recipe scores higher (GRU4Rec 0.3967, SASRec 0.3856, BERT4Rec 0.3460, SASRecF 0.3997, FDSA 0.3949, DIF-SR 0.4177), under which our model reaches only 89.5% of the strongest; re-scoring those original checkpoints on corrected targets moves them heterogeneously (GRU4Rec 0.3281, SASRecF 0.3707, SASRec *rises* to 0.3887), while a control re-score of our own model measures +0.0065 the other way, a small pure-easiness term. The audit procedure is deterministic (§Reproducibility). **Protocol sensitivity.** Echo is universal across the Table 1 families (baseline roster 3.1–4.2 seen items in the top 20; RetrievalFormer 4.42 at seen-window parity) and is a protocol lever: Masking lifts every model by +0.030 to +0.061 Recall@20, and masking *both* sides of a pair closes at most about half of its unmasked gap (17 gated two-pass rescores, five benchmarks). Protocol choice moves levels, not rankings: no ordering changes at any masking depth we tested, and masking only each user's five most recent items already recovers 35–50% of the full-window gain. Mechanism and cross-benchmark normalization are in Appendix B.

Cross-benchmark generalization check. The warm gap is benchmark-dependent. On a MIND-large 5-core next-item benchmark (267,545 users; 9,353 items; mean history ∼11 events, median 8, so the history-cap lever of RQ2(b) is structurally absent and our arms' cap of 100 against the roster's 50 is inert; every model ranks with an exact full-catalog softmax, BERT4Rec's cloze-style and DIF-SR adding its published auxiliaries), RetrievalFormer scores Recall@20 0.3368 ± 0.0017 (3 seeds; top seed 0.3387) against the same own-run, tuned RecBole roster (Table 2): within 0.8–3.6% of the five strongest, the tightest band of any benchmark we measure, though sixth of seven rows, while exceeding BERT4Rec (0.3275±0.0055). On AliEC (30,461 users; 16,475 items after alignment), where our model and the roster both consume a 50-item history window, it scores 0.0786 against a leaderboard that tuning kept reordering (top configurations FDSA at 0.0927 ± 0.0034 and DIF-SR at 0.0927 ± 0.0004, statistically tied; the top spot changed twice across the rounds). MIND-large's tight band is structural: short histories remove the largest single axis of the MovieLens gap, and scoring parity holds on both printed benchmarks (the memory wall of §4.3(d) was measured on the separate raw 240K-item AliEC catalog, for which no head-to-head row is printed).

Table 2: MIND-large 5-core next-item benchmark (267,545 users; 9,353 items): full-catalog unmasked Recall@20, own runs under the same retrain-and-tune protocol as Table 1, with the same genre-consuming inputs. Mean±std over 3 seeds; FDSA is a single run. Short histories (mean ∼11 events, median 8) remove the history-context axis of §4.3(b), and every model ranks with an exact full-catalog softmax rather than a sampled one. Best per column bold, runner-up underlined.

Model	Recall@20	Seeds
DIF-SR	**0.3494±0.0011**	3
FDSA	<u>0.3446</u>	1
SASRec	0.3433±0.0008	3
SASRecF	0.3430±0.0003	3
GRU4Rec	0.3396±0.0016	3
BERT4Rec	0.3275±0.0055	3
RetrievalFormer	0.3368±0.0017	3

4.3 RQ2: Where the Gap Comes From

(a) The training recipe is the largest measured contributor. The recipe swap moves the score more than any other paired lever. On MIND-small, in a single matched pair with architecture, features, data, and budget held identical, switching from sampled InfoNCE (96 mixed negatives per example: 48 in-batch, 24 uniform, 14 popular, 10 tail; cosine scoring, no log-Q correction) to full-softmax cross-entropy (dot-product scoring with a learned item bias) moves test Recall@20 from 0.1830 to 0.2822 (+54% relative). Exact-convention two-pass rescores validate both arms as unmasked-equivalent to within 0.0006; loss, scoring, sampling, and schedule move together as deployed on this leg. On MovieLens-1M at a matched earlier configuration (4 layers, one pair), the same swap lifts test Recall@20 from 0.3132 to 0.3349 (+6.9% relative; NDCG@20 +8.9%). Both arms sit at a plateau: a converged relaunch early-stopped at the identical checkpoint. The MovieLens-1M anatomy is explicit. The dissection runs from the matched sampled-objective configuration (0.3132) to SASRec's legacy realized-target value (0.3856), the endpoint that framed this dissection before the target correction of §4.2: a distance of 0.0724. Of that, the measured recipe contribution recovers +0.0217, configuration sweeping adds +0.0022, and 0.0485 remains as the residual of (b). Against the corrected benchmark's strongest baseline (DIF-SR, 0.3944) the distance is 0.0812 and the same legs attribute 0.0239 of it, so the unattributed remainder is bracketed between 0.0117 and 0.0205 depending on which endpoint anchors the anatomy; we quote the wider figure where the corrected roster is the comparison. The objective is thus the largest single lever we can isolate by a *paired* experiment; the history-context axis of (b), attributed by sweep rather than pairing, moves more (+0.0297 vs. +0.0217), half the post-recipe remainder against the corrected endpoint; neither is the majority of the gap on this benchmark. Sampled training starves the ordering signal: a given competitor receives gradient with probability ≈0.04 per query where full-softmax CE updates all 3,415 every step, and NDCG improves proportionally more than recall in both pairs (mechanism, prior art, and two caveats on sampled recipes: Appendix C).

(b) A residual survives the axes we swept. With the full-softmax objective and each swept axis (schedule, dropout, learning rate, batch size, capacity, depth) at its best value at the earlier cap-100 configuration, RetrievalFormer's 0.3371 trails the legacy-target SASRec by 0.0485 (12.6% relative). Measurements bound what this residual is *not*, within the tested bounds: not an evaluation-protocol artifact (masking both sides closes only ∼44% of the checkpoint-pair gap, §4.2), not approximate search (below), not embedding sharing (within seed noise at the optimum; ablation below), and not supervision density. On that last axis, a screen registered before the run replaced the objective with per-position full-softmax cross-entropy on the same architecture and initialization and *lowered* test Recall@20 from 0.3684 to 0.3260 (single seed, but 7× the ±0.006 screen threshold). A first direct probe also finds no effect from the item-ID pathway: a single-seed variant trained without the user-side item-ID input or its tied output-side component lands within seed noise of that configuration (0.3373 vs. 0.3371). It is *consistent with* the class difference on the item side: the ID-softmax model's item representation is a free per-item parameter vector (pure memorization capacity, updated for every item at every step under full-softmax CE), while ours is constrained to a feature function, which buys zero-shot cold items (§4.4). Dual-encoder capacity analyses in dense text retrieval reach similar conclusions [22], and richer accelerator-friendly similarity heads recover expressiveness within the retrieval envelope [36, 53]. The history-context axis attributes most of it: the earlier pinned configuration caps history at 100 items, covering the median user (median history 96) but discarding 55% of training interactions; raising the cap to 300 (recovering 78.5%) lifts the same architecture and recipe to 0.3668 ± 0.004 Recall@20 (5 seeds; per-seed range 0.3624–0.3725), with cap 200 (0.3589 ± 0.002, 2 seeds) tracking the coverage curve and cap 400 saturating (0.3675, single seed). How much of that window the model needs at *inference* is separable, and smaller: re-scoring the submitted checkpoints with evaluation history truncated to the baselines' 50 items (padded width held fixed so retained items keep their trained positions) costs 0.3739 → 0.3671 ± 0.0023 (3 seeds, paired, every seed negative). At that context parity the warm gap widens from −0.0205 to −0.0273 against DIF-SR (93.1% of its bar, against 94.8% printed), and our ordering against FDSA resolves against us (Welch t=3.4) where at full window it does not (t=1.1); against GRU4Rec the two remain indistinguishable at either window (t=0.6). This caps our model against baselines at their native cap, the fair direction rather than a like-for-like recompute. Roughly three-quarters of the cap's value is therefore recovered training interactions, not inference context. That mechanism is MovieLens-specific: we truncate at ingest, so an interaction outside the window enters no training row, and at each bed's own cap this discards 21.5% of MovieLens-1M interactions against 0.1% on MIND (mean history 10.7) and 0.2% on AliEC (mean 8.3), where the window is inert and the asymmetry with the roster's 50 costs nothing. Splitting the tower learning rates on that configuration (user 3×10^{-4}, item 1×10^{-3}, cosine schedule) adds +0.007, reaching the submitted 0.3739 ± 0.0013 (3 seeds, GPU-verified exact rescores) of Table 1. On the corrected benchmark (Table 1) the submitted model sits above the ID-only GRU4Rec and BERT4Rec and 0.003–0.021 under the four strongest (DIF-SR the farthest); measured against the legacy realized-target SASRec value that framed this dissection, the unattributed remainder is 0.012; loss variants and supervision density remain unexhausted. A cutoff decomposition localizes it: against the strongest baseline (DIF-SR) the ratio is monotone in the cutoff: 86% at Recall@5, 92% at Recall@10, 95% at Recall@20, and seen-item exclusion on both sides closes roughly half of that head gap (86%→92% at Recall@5), so the head-of-list deficit is part evaluation echo and part the item-side memorization account above (Appendix B). Isolating the remainder (further ID-pathway factorization, distillation from candidate-conditioned teachers) is future work.

(c) ANN approximation error contributes nothing to the accuracy gap. Every accuracy number here comes from exact, exhaustive scoring of its full candidate set, so the informal attribution of dual-encoder gaps to "the ANN approximation" is ruled out by construction, and a real-embedding check agrees: HNSW on the MIND-large checkpoint reproduces the exact end metric within noise (§4.5).

(d) The recoverable leg hits a memory wall at scale. The exact objective stops fitting in memory before the catalog gets large. The full-softmax objective materializes the entire item tower forward (with gradients) every training step, so its memory footprint carries a term linear in the *catalog* that batch reduction cannot remove: in our implementation on a single 44 GB GPU it trains at a 3.4K-item catalog (MovieLens-1M) and a 9K-item catalog (MIND) but goes out of memory at a 240K-item catalog at both tested batch sizes (512 and 256); the failing 470 MiB allocation matches one catalog-wide d=512 fp32 activation. The constraint is not a kernel artifact. For a plain ID-parameterized output layer, fused cross-entropy kernels eliminate the logit memory outright [44]; but a *feature-based* item tower must recompute the catalog-wide tower forward and backpropagate through it every step, so even memory-optimal exact training pays per-step compute linear in catalog size (we did not test gradient checkpointing, activation offload, or sharding, which relax memory but not this compute term), and the practical escapes buy scale by leaving the exact objective behind: caches score against stale embeddings [20, 41] and adaptive-softmax factorizations restructure the loss [6]. Prior work reaches the same verdict from the ID-softmax side (full softmax impractical at million-item catalogs [27]); concurrent work engineers around precisely this wall with approximate cross-entropy losses [8, 24]. At a 2.2M-item catalog a full d=512 table is 4.2 GiB of parameters and 16.8 GiB with gradients and Adam moments, which is why million-item work in this line represents items by *compressed* sub-item embeddings rather than a full per-item table [29]. Both classes pay a catalog-linear training cost (theirs in item parameters, ours in the tower forward), so the wall is a property of exact catalog-scale objectives. The exact objective that anchors the recoverable part of the gap is trainable where the dual encoder is least needed (small catalogs) and infeasible at the scales that motivate it. Closing the distance between corrected sampled objectives [16, 17, 47] and the full-softmax anchor at catalog scale is an open problem for retrieval-based sequential recommendation.

Within-class ablations (Appendix C). Sharing embedding tables across towers matters on average but not at the optimum (marginalized over the tuning grid: separate tables cost 3.4% Recall@20, no shared features 6.9%; within seed noise at the best configuration), robustness, not an accuracy driver. Replacing prefix-LM attention with pure causal attention costs −0.0098/−0.0026 on

Table 3: Strict item cold-start on MovieLens-1M (global random 20% item holdout; cold/warm slices rank their candidate partitions, overall ranks the full catalog). Mean±std over 3 seeds where shown; row definitions and protocol in §4.4 and Appendix A. MF and the dedicated baselines are tuned through five documented one-lever rounds with 3-seed champion confirms. Item content is frozen sentence embeddings ("text") or genre+year ("meta"); text variants remain at reference configurations. Best per column by point estimate bold, runner-up underlined; "—" = not computed. The three bolded warm cells are the same tuned-MF ceiling, which the fused rows inherit by construction.

Model	Cold R@20	Warm R@20	Overall
Floors			
Last-item text cosine (no training)	0.058	0.018	—
MF (ID-based CF, tuned)	0.031±0.004	**0.276±0.001**	**0.125**
Dedicated cold-start baselines (tuned)			
DropoutNet (meta)	0.094±0.013	0.241±0.001	0.109
Heater (meta)	0.106±0.005	0.256±0.002	0.116
ALDI	0.124±0.007	0.267±0.000	0.121
RetrievalFormer: varying how a warm channel is attached			
Content tower, no warm channel	0.172±0.006‡	0.113±0.005	0.067
+ warm-masked CE (retrained)	0.154±0.003	0.131±0.005	—
⊕ tuned MF (score fusion)	0.172±0.006‡	**0.276±0.001**	**0.125**±0.000
⊕ tuned MF, instr. encoder	**0.195±0.008**	**0.276±0.001**	**0.125**±0.000

‡These two cold values are equal by construction: MF's cold rows are zeroed and the fusion scalar is cold-slice invariant, so the same frozen content tower is the sole cold scorer in both rows. Only the warm channel differs.

paired seeds, so the bidirectional prefix block is load-bearing at the margin and the submitted configuration keeps it.

4.4 RQ3: Strict Item Cold-Start

Protocol. We use a strict item cold-start split with zero leakage: 20% of items (742 of 3,706; grouped by item, seed-fixed) are held out as cold, and *every* interaction involving a cold item is removed from training. This benchmark retains all rated MovieLens-1M items; the warm-accuracy benchmark of §4.2 is 5-core filtered (3,416). Cold items are split into disjoint validation and test sets (371 each); warm interactions are split 8:1:1. Following the benchmark's slice convention, the cold slice ranks all 742 cold items, the warm slice ranks all 2,964 warm items, and the overall slice ranks the full catalog, always excluding the user's training items; cold items are candidates for every model, so the cold-start tax is explicit. A global random item holdout samples the catalog uniformly, unlike per-user leave-last-out; exhaustive partition ranking, unlike sampled-negative AUC, is comparable to the strict cold-start literature [11, 40, 43, 57]. Appendix A details splits and baseline training. Each baseline is retrained to convergence at 3 seeds and tuned through the same five-round, one-lever protocol as the warm baselines (§4.1). Analytic floors: a random ranker's cold Recall@20 is $20/742 \approx 0.027$, and a popularity ranker is uninformative on cold items (all have zero training interactions). We add a behavioral floor: ranking candidates by cosine similarity between the frozen text embedding of the user's last training item and each candidate's embedding (zero training, byte-identical frozen features to our item tower), reaches 0.058 cold Recall@20, a floor any *trained* cold-start method should clear.

Results (Table 3). The ID-based CF model collapses on cold items (0.031 ± 0.004, within noise of the 0.027 chance floor). The dedicated cold-start baselines clear the CF floor, and tuning moves them substantially (Heater 0.087 → 0.106, DropoutNet 0.077 → 0.094; ALDI reaches 0.124), but DropoutNet's and Heater's *text* variants land at the training-free last-item floor (0.054–0.059 vs. 0.058; Appendix C): strong content alone does not suffice. RetrievalFormer's content-only tower reaches 0.172 ± 0.006 cold (3 seeds): 1.4× ALDI, 1.6× tuned Heater, 3× the floor, 5.5× tuned MF, from the retrieval architecture without a dedicated cold-start training stage. These margins are over the three methods we could retrain; we omit CLCRec (§5). The warm-masked full-softmax variant (the same recipe with cold items masked out of the training softmax, so the loss trains on warm targets only; 3 seeds) trades cold for warm (0.154 ± 0.003 / 0.131 ± 0.005), consistent with RQ2(a): the recipe arc from sampled InfoNCE lifts warm from ~0.07 to 0.131 at little cold cost, closing under a third of the deficit to the tuned-MF ceiling, and raising the learning rate toward its warm optimum monotonically degrades cold retrieval. An encoder swap is not automatically the lever: substituting a newer, MTEB-stronger 0.6B text-embedding model into the champion recipe unchanged *lowers* cold recall by 19% relative — a statement about reusing a recipe tuned for one embedding geometry, not about encoder quality, since we did not re-tune under the new encoder.

A fused cold/warm operating point. A frozen score fusion reaches both axes at once. Fusing the frozen content encoder with the frozen *tuned* MF of Table 3 (zeroing MF's cold rows; one scalar weight per content seed, selected on warm-slice validation; the cold slice is s-invariant by construction) attains cold 0.172 ± 0.006 / warm 0.276 ± 0.001 (3 seeds): by construction cold ranking is carried entirely by the content component and warm saturates at the backbone's level, so the fusion carries the content tower's cold at the tuned-MF warm ceiling, above every dedicated baseline's 3-seed row on both axes. The overall slice is 80% warm items, so it tracks warm capacity: the fusion lifts the content tower from 0.067 to 0.125 there, the best overall value in the table. Attaching a frozen CF backbone to a content encoder is the construction the dedicated baselines use (DropoutNet, Heater, and ALDI are all built on one); the fusion differs in combining at score level rather than in a learned layer, and in requiring no additional training. The fusion holds across content models. With an instruction-tuned encoder variant, the same protocol reaches cold 0.195 ± 0.008 at the same warm ceiling (0.276 ± 0.001; 3 seeds, one test read at the validation-selected weight), the strongest fused point we measure. Two caveats: the win-both operating point is the fusion (the DropoutNet/Heater recipe with a stronger content encoder and a properly trained CF backbone), and *co-training* a single model to win both did not, across five single-seed hybrid arms, though objective isolation does not require separate serving models. We train the memorization (ID-embedding CF) channel in isolation and select its endpoint on validation: a 3-point training-budget curve whose later endpoints overfit the memorization table, ranked correctly by validation, so the choice involves no test peeking. We then *graft* the content champion's weights under it; the two parameter groups are disjoint, blended by the same validation-selected scalar protocol as the fusion. The result is a single valid checkpoint at cold 0.172 ± 0.006 / warm 0.250 ± 0.000 (3 content seeds under the same

channel): cold fully held, warm at 90% of the tuned-MF ceiling, zero additional training. A sanity gate confirms the grafted content path reproduces the champion's scores per seed. Co-training did not win both; training the channels separately and composing them does. This composition carries two-run provenance.

Two controls (Appendix C). Giving DropoutNet and Heater the same frozen text embeddings *drops* their cold recall (Heater 0.087 → 0.054), so the advantage tracks how the architecture uses the encoder, not the encoder itself; and the warm deficit is protocol-relative, each class winning its matched task (0.267 vs. our 0.113 on matrix completion, 0.350 vs. MF's 0.125 on leave-one-out). Only the feature-based tower also wins cold.

Cold coverage on the search surface. The openness premise is a search-side claim, so we test it there, on the Avito context-ad log: query-initiated search (112.2M searches, 15.6% with free-text queries) and organic browsing (286.8M events) over one 36.9M-ad catalog. Freshness pressure is heavy independent of any label choice — of 9.77M query-search impressions in a held-out week, 38.6% show an ad absent from every earlier impression *and* visit (41.0% of distinct ads shown), the premise's antecedent measured rather than assumed. Organic inventory carries no click labels, so relevance comes from delayed cross-stream engagement: a shown ad the same user visits 30 minutes to 24 hours later (246,443 pairs; the floor discards the 81.9% that are SERP click-throughs, median gap 28 s, and a placebo pairing each search with another user's impressions converts 1,700× less often). Against 99 sampled negatives, on engagements where the user had no prior contact with the item, so only content can retrieve it, Recall@20 is 0.9595±0.0008 cold against 0.9510 ± 0.0008 warm (3 seeds, 12,946/15,368 instances; query tower alone 0.9510/0.9433). Two receipts bound that reading. Under the same sampled protocol, untuned BM25 over unstemmed titles also ranks cold above warm (0.7596/0.7261 pooled, against the tower's 0.9685/0.9646), so the slice favors cold items independently of the tower; and on the undivided test set the strongest model-free scorer we built reaches 0.8791 against the tower's 0.9664, so the sampled task is easy but not saturated. And against the entire 22.8M-ad pool (single seed; 549/638 of a 2,000-instance sample fall in this stratum) cold and warm are indistinguishable at every depth (−0.71pp Recall@20 [−1.83, +0.31]; +1.86pp Recall@1000 [−1.74, +5.46]). The claim is the absence of a *measurable* cold penalty, not a cold advantage, and not parity at catalog scale.

4.5 RQ4: Serving as a Cost Ledger

Exact full-catalog scoring of dual-encoder embeddings holds 90th-percentile (p90) latency under 30 ms up to 1M items, so approximation is unnecessary at this scale. Within this envelope (1,024-query batches over synthetic d=512 vectors, wider than our d_{model}=256, so the scan term is conservative), every accuracy benchmark here is scored exactly, without approximation. Beyond that, exact scan reaches 292 ms at 10M items against an aggressive IVF-PQ index's ∼1 ms, which retrieves only 2.8% of the exact top-K (m=16, n_{probe}=16). A real-embedding check closes the loop between index recall and the end metric: on the certified MIND-large checkpoint (9,353 items, 267,545 test users; the learned item bias folded in by coordinate augmentation, so index inner product equals the exact score), HNSW (M=32, depth 100) reproduces exact-scan Recall@20 within noise (0.33878 vs. 0.33884, 99.98% retention) and IVF-Flat retains 99.86%, but at this scale ANN is not a latency win: exact CPU scan is already 0.28 ms p90 per query, and the user-tower forward (∼5 ms) dominates regardless of retriever. Approximation becomes load-bearing only above $\sim 10^6$ items, where the batched scan reaches tens of milliseconds per query; fidelity can then be bought back (raising n_{probe} from 8 to 16 lifts recall from 0.021 to 0.028 for +0.1 ms at 10M items). Serving cost is symmetric across the classes, and scale-dependent. Timing the retrained SASRec champion's own scoring path (single-thread CPU, d=256) puts the sequence forward at 94–99.5% of its *full_sort_predict* time (encoder 2.2–3.0 ms over that call's 2.3–3.0 ms, batch sizes 1 and 1024), with the scan timed in isolation at 0.02–0.09 ms, at this benchmark's 3,416 items plus the padding row; the same scan on synthetic catalogs inverts that ratio, the encoder falling to 27% of cost at 10^6 items and 4% at 10^7. Both classes run one sequence forward and one scan against a static table, so the profile is theirs equally: at benchmark scale neither is scan-bound, at catalog scale both are. End-to-end ANN impact at 10^6-plus scale is future work.

5 Limitations

Warm-accuracy and cold-start grids are on MovieLens-1M; MIND and AliEC enter as paired-objective and cross-benchmark checks (mixed single-run and 3-seed rows, Table 2). The cross-protocol MF cell is single-seed (the fusion, grafted composition, and warm-masked CE rows are 3-seed; the graft carries two-run provenance); baselines in Table 1 are 3-seed means of each champion configuration, and the cold-start baselines are 3-seed and tuned through the same round protocol (text variants remain at reference configurations). We did not enforce parameter parity, and our model's configuration search was broader than the five rounds each baseline received, a budget asymmetry favoring our model. The baselines also run their framework's default 50-item history window, and history length was not among their five tuning levers, so the history-context leg of §4.3(b) is an axis our model was swept on and they were not. We did not retrain a MovieLens baseline at a longer window, so on that benchmark we cannot separate that leg from what a baseline would gain from the same window — only what our own model loses without it (§4.3); on AliEC, where both sides run the 50-item window natively, we trail the two strongest by 15%. The dedicated cold-start roster is three methods rather than four: we omit CLCRec, whose available implementations we could not clear under licensing review, and cite it as prior art only. The residual in RQ2(b) is bounded by elimination and largely attributed by the history-context axis (the submitted configuration is the 3-seed split-LR cap-300 family; the dissection used the 5-seed constant-LR cap-300 family), and the remaining isolations (supervision density, loss variants) have not run. Scan timings beyond 10^4 items use synthetic embeddings on one host and vary by ∼15% run to run, and absolute latencies are hardware-dependent. The batched benchmark measures the catalog-scan term only, and the quoted IVF-PQ point is an aggressive-quantization, low-recall configuration. The ANN sweep covers one index family at two probe settings and the real-embedding end-metric check one benchmark (MIND-large); end-metric ANN impact at 10^6-plus-item scale remains unmeasured. The objective effect of §4.3(a) is a recipe bundle, and quality

and trainability were not measured across catalog scales on one benchmark. No head-to-head accuracy comparison runs at the 10^5-plus-item scale that motivates the architecture: the accuracy grids are 3.4K and 9.4K items, and the memory wall is measured at 240K without an accompanying accuracy read. A feature-based tower also inherits metadata-quality bias: items with sparse or malformed descriptions receive lower-quality embeddings, a coverage concern for long-tail segments that our method does not correct. The search-side result pairs a label-free prevalence count with a cold-coverage read whose relevance signal is delayed engagement on one external log, and whose lexical comparator is an untuned BM25 rather than a tuned search baseline. Its full-catalog leg is a single model seed, and the 2,000-instance sample leaves 1,187 in the no-prior-contact stratum, so its Recall@20 interval spans about 2pp: it establishes that no cold penalty is measurable at that power, not that none exists. Training the second tower as a query encoder and measuring blended retrieval against a tuned baseline is future work.

6 Conclusion

The exact objective is trainable where the architecture is least needed, and infeasible (in our implementation) at the catalog scales a shared index demands. That trade-off is the central finding. The cost is 94.8% of the strongest retrained baseline, on targets a float32 artifact had been silently reordering (§4.2); in exchange the index stays open, at 1.4× the strongest of the three dedicated methods we retrained, with no retrieval-stage cost regression (§4.4). The memory wall remains open.

References

[1] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.

[2] Gabriel de Souza Pereira Moreira, Sara Rabhi, Jeong Min Lee, Ronay Ak, and Even Oldridge. 2021. Transformers4Rec: Bridging the gap between NLP and sequential/session-based recommendation. In *Proceedings of the 15th ACM Conference on Recommender Systems*. 143–153.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*. 4171–4186.

[4] Chantat Eksombatchai, Pranav Jindal, Jerry Zitao Liu, Yuchen Liu, Rahul Sharma, Charles Sugnet, Mark Ulrich, and Jure Leskovec. 2018. Pixie: A system for recommending 3+ billion items to 200+ million users in real-time. In *Proceedings of the 2018 world wide web conference*. 1775–1784.

[5] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. 2019. Are we really making much progress? A worrying analysis of recent neural recommendation approaches. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 101–109.

[6] Edouard Grave, Armand Joulin, Moustapha Cissé, David Grangier, and Hervé Jégou. 2017. Efficient softmax approximation for GPUs. In *Proceedings of the 34th International Conference on Machine Learning*. PMLR, 1302–1310.

[7] Mihajlo Grbovic and Haibin Cheng. 2018. Real-time personalization using embeddings for search ranking at airbnb. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 311–320.

[8] Danil Gusak, Gleb Mezentsev, Ivan V. Oseledets, and Evgeny Frolov. 2024. RECE: Reduced Cross-Entropy Loss for Large-Catalogue Sequential Recommenders. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 3772–3776. doi:10.1145/3627673.3679986

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

[10] Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. 2022. Towards Universal Sequence Representation Learning for Recommender Systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 585–593.

[11] Feiran Huang, Zefan Wang, Xiao Huang, Yufeng Qian, Zhetao Li, and Hao Chen. 2023. Aligning distillation for cold-start item recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1147–1157.

[12] Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronber, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. 2020. Embedding-based Retrieval in Facebook Search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2553–2561.

[13] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data* 7, 3 (2019), 535–547.

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

[15] Kirill Khrylchenko, Vladimir Baikalov, Sergei S. Makeev, Artem Matveev, and Sergei Liamaev. 2025. Correcting the LogQ Correction: Revisiting Sampled Softmax for Large-Scale Retrieval. In *Proceedings of the 19th ACM Conference on Recommender Systems*. 545–550. doi:10.1145/3705328.3748033

[16] Anton Klenitskiy and Alexey Vasilev. 2023. Turning Dross Into Gold Loss: is BERT4Rec really better than SASRec?. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 1120–1125. doi:10.1145/3604915.3610644

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

[18] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. 2019. Set Transformer: A Framework for Attention-based Permutation-Invariant Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 3744–3753.

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

[20] Erik Lindgren, Sashank J. Reddi, Ruiqi Guo, and Sanjiv Kumar. 2021. Efficient Training of Retrieval Models using Negative Cache. In *Advances in Neural Information Processing Systems*, Vol. 34. 4134–4146.

[21] Gang Liu, Fan Yang, Yang Jiao, Alireza Bagheri Garakani, Tian Tong, Yan Gao, and Meng Jiang. 2025. Learning Attribute as Explicit Relation for Sequential Recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*. 800–811. doi:10.1145/3690624.3709267

[22] Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, Dense, and Attentional Representations for Text Retrieval. *Transactions of the Association for Computational Linguistics* 9 (2021), 329–345. doi:10.1162/tacl_a_00369

[23] Yu A Malkov and Dmitry A Yashunin. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence* 42, 4 (2018), 824–836.

[24] Gleb Mezentsev, Danil Gusak, Ivan V. Oseledets, and Evgeny Frolov. 2024. Scalable Cross-Entropy Loss for Sequential Recommendations with Large Item Catalogs. In *Proceedings of the 18th ACM Conference on Recommender Systems*. 475–485. doi:10.1145/3640457.3688140

[25] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[26] Nikil Pancha, Andrew Zhai, Jure Leskovec, and Charles Rosenberg. 2022. PinnerFormer: Sequence Modeling for User Representation at Pinterest. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3702–3712. doi:10.1145/3534678.3539156

[27] Aleksandr V. Petrov and Craig Macdonald. 2023. gSASRec: Reducing Overconfidence in Sequential Recommendation Trained with Negative Sampling. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 116–128. doi:10.1145/3604915.3608783

[28] Aleksandr V. Petrov, Craig Macdonald, and Nicola Tonellotto. 2024. Efficient Inference of Sub-Item Id-based Sequential Recommendation Models with Millions of Items. In *Proceedings of the 18th ACM Conference on Recommender Systems*. 912–917. doi:10.1145/3640457.3688168

[29] Aleksandr V. Petrov, Craig Macdonald, and Nicola Tonellotto. 2025. Efficient Recommendation with Millions of Items by Dynamic Pruning of Sub-Item Embeddings. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2102–2111. doi:10.1145/3726302.3729963

[30] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan H. Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Maheswaran Sathiamoorthy. 2023. Recommender Systems with Generative Retrieval. In *Advances in Neural Information Processing Systems*, Vol. 36.

[31] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. 2002. Methods and Metrics for Cold-Start Recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 253–260.

[32] Teng Shi, Zihua Si, Jun Xu, Xiao Zhang, Xiaoxue Zang, Kai Zheng, Dewei Leng, Yanan Niu, and Yang Song. 2024. UniSAR: Modeling User Transition Behaviors between Search and Recommendation. In *Proceedings of the 47th International*

ACM SIGIR Conference on Research and Development in Information Retrieval. doi:10.1145/3626772.3657811

[33] Zihua Si, Zhongxiang Sun, Xiao Zhang, Jun Xu, Xiaoxue Zang, Yang Song, Kun Gai, and Ji-Rong Wen. 2023. When Search Meets Recommendation: Learning Disentangled Search Representation for Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. doi:10.1145/3539618.3591786

[34] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. AutoInt: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1161–1170.

[35] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.

[36] Liangcai Su, Fan Yan, Jieming Zhu, Xi Xiao, Haoyi Duan, Zhou Zhao, Zhenhua Dong, and Ruiming Tang. 2023. Beyond Two-Tower Matching: Learning Sparse Retrievable Cross-Interactions for Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1288–1297.

[37] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.

[38] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 565–573.

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

[40] Maksims Volkovs, Guangwei Yu, and Tomi Poutanen. 2017. DropoutNet: Addressing cold start in recommender systems. In *Advances in neural information processing systems*, Vol. 30. 4958–4968.

[41] Jinpeng Wang, Jieming Zhu, and Xiuqiang He. 2021. Cross-Batch Negative Sampling for Training Two-Tower Recommenders. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1632–1636. doi:10.1145/3404835.3463032

[42] Tongzhou Wang and Phillip Isola. 2020. Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere. In *Proceedings of the 37th International Conference on Machine Learning*. PMLR, 9929–9939.

[43] Yinwei Wei, Xiang Wang, Qi Li, Liqiang Nie, Yan Li, Xuanping Li, and Tat-Seng Chua. 2021. Contrastive learning for cold-start recommendation. In *Proceedings of the 29th ACM International Conference on Multimedia*. 5382–5390.

[44] Erik Wijmans, Brody Huval, Alexander Hertzberg, Vladlen Koltun, and Philipp Krähenbühl. 2025. Cut Your Losses in Large-Vocabulary Language Models. In *The Thirteenth International Conference on Learning Representations*.

[45] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. 2017. Recurrent recommender networks. In *Proceedings of the tenth ACM international conference on web search and data mining*. 495–503.

[46] Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, and Ming Zhou. 2020. MIND: A large-scale dataset for news recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 3597–3606.

[47] Jiancan Wu, Xiang Wang, Xingyu Gao, Jiawei Chen, Hongcheng Fu, and Tianyu Qiu. 2024. On the Effectiveness of Sampled Softmax Loss for Item Recommendation. *ACM Transactions on Information Systems* 42, 4 (2024), 98:1–98:26. doi:10.1145/3637061

[48] Yueqi Xie, Peilin Zhou, and Sunghun Kim. 2022. Decoupled Side Information Fusion for Sequential Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1611–1621. doi:10.1145/3477495.3531963

[49] Ji Yang, Xinyang Yi, Derek Zhiyuan Cheng, Lichan Hong, Yang Li, Simon Xiaoming Wang, Taibai Xu, and Ed H Chi. 2020. Mixed negative sampling for learning two-tower neural networks in recommendations. In *Companion Proceedings of the Web Conference 2020*. 441–447.

[50] Liu Yang, Fabian Paischer, Kaveh Hassani, Jiacheng Li, Shuai Shao, Zhang Gabriel Li, Yun He, Xue Feng, Nima Noorshams, Sem Park, Bo Long, Robert Nowak, Xiaoli Gao, and Hamid Eghbalzadeh. 2024. Unifying Generative and Dense Retrieval for Sequential Recommendation. *arXiv preprint arXiv:2411.18814* (2024).

[51] Xinyang Yi, Ji Yang, Lichan Hong, Derek Zhiyuan Cheng, Lukasz Heldt, Aditee Kumthekar, Zhe Zhao, Li Wei, and Ed Chi. 2019. Sampling-bias-corrected neural modeling for large corpus item recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 269–277.

[52] Hamed Zamani and W. Bruce Croft. 2020. Learning a Joint Search and Recommendation Model from User-Item Interactions. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. doi:10.1145/3336191.3371818

[53] Jiaqi Zhai, Zhaojie Gong, Yueming Wang, Xiao Sun, Zheng Yan, Fu Li, and Xing Liu. 2023. Revisiting Neural Retrieval on Accelerators. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 5520–5531. doi:10.1145/3580305.3599897

[54] Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S. Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, and Xiaofang Zhou. 2019. Feature-level Deeper Self-Attention Network for Sequential Recommendation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*. 4320–4326. doi:10.24963/ijcai.2019/600

[55] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. 2021. RecBole: Towards a Unified, Comprehensive and Efficient Framework for Recommendation Algorithms. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*. 4653–4664. doi:10.1145/3459637.3482016

[56] Kun Zhou, Hui Yu, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Filter-enhanced MLP is all you need for sequential recommendation. In *Proceedings of the ACM Web Conference 2022*. 2388–2399.

[57] Ziwei Zhu, Shahin Sefati, Parsa Saadatpanah, and James Caverlee. 2020. Recommendation for new users and new items via randomized training and mixture-of-experts transformation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1121–1130.

Reproducibility Statement

All datasets are public (MovieLens-1M, MIND, AliEC). Item text is encoded by a frozen public sentence encoder (gte-Qwen2-1.5B, 1,536-dim, L2-normalized); the cold-start encoder-substitution control of §4.4 swaps it for a newer 0.6B model at higher MTEB rank. Baselines are retrained in RecBole with full-softmax cross-entropy, tuned through five documented one-lever rounds each, and evaluated full-catalog unmasked. Cross-harness data and candidate parity was verified directly (identical 5-core statistics and train-interaction counts; a popularity ranker scores 0.0402 unmasked Recall@20 in both stacks), and build gates assert target identity for all 6,040 users on both corrected splits. Large language models were used for literature search and writing assistance only; every reported number originates from a logged training or evaluation run.

A Cold-Start Protocol Details

Interactions are grouped by item and items are shuffled with a fixed seed; 80% of items (with all their interactions) form the warm pool, split 8:1:1 into train/validation/test by interaction. The remaining 20% of items contribute *all* of their interactions to the cold pool, split 1:1 by item into disjoint cold-validation and cold-test sets, so no cold item's ID or interactions appear in training and no item is shared between cold validation and cold test. This benchmark is immune by construction to the timestamp-quantization artifact of §4.2: its targets are item-random rather than time-derived (the split procedure contains no timestamp logic), and our pipeline preserves integer timestamps end-to-end (verified over all 646,073 events). At evaluation, each slice ranks its candidate partition minus the user's training history: the cold slice all 742 cold items, the warm slice all 2,964 warm items, and the overall slice the full catalog, with cold items scored by every model; ID-based models score cold items through whatever mechanism they possess (for MF, the untrained embedding). Cold-start baselines (DropoutNet, Heater, ALDI) are trained on the warm pool with item content (genre+year metadata, or frozen sentence embeddings of titles in the "text" variants) and evaluated identically. All three consume the same frozen matrix-factorization backbone. Each ran a published-config retrain plus five

documented one-lever tuning rounds (validation-selected; champions confirmed at 3 seeds), under a 500-epoch cap with early stopping; text variants remain at reference configurations (e.g., Heater's 200-unit expert MLP), without per-variant capacity re-tuning. The last-item floor scores every candidate by cosine similarity to the frozen text embedding of the user's most recent training item, using the same frozen embeddings our item tower consumes. Model selection for all cold models uses cold-validation; the fusion scalar s in Table 3 is selected to maximize warm-slice validation accuracy (under a cold floor that never binds) and applied once to test; the cold slice is invariant to s by construction, since MF's cold rows are zeroed and the cold candidate partition contains only cold items.

B Evaluation-Protocol Sensitivity

Echo mechanism: on single-consumption benchmarks like MovieLens a seen item can never be the held-out target, so an evaluation-time seen-item filter converts every echoed slot into free rank improvement, hence the paper's unmasked numbers plus the echo diagnostic. The RetrievalFormer echo of §4.2 is counted at seen-window parity on the printed history-300 seeds and sits above the top of the baseline band. Nor is echo a MovieLens quirk: on the sparser MIND and AliEC benchmarks, echo runs 15–150× the random-ranking expectation (20× mean history length / catalog size), and masking flips no model comparison. The ∼44% gap closure quoted in §4.3(b) is the SASRec pair under both-sides masking. Echo is computed from input sequences alone and is independent of the target-definition artifact of §4.2.

Cutoff decomposition (§4.3(b)): the submitted model's NDCG ratios against DIF-SR are 85%/89%/91% at cutoffs 5/10/20, and both-sides seen-item exclusion lifts the NDCG@5 ratio 85%→90%. Our side is 3 seeds; the baseline side is single-seed re-scores of the champion checkpoint, so the @20 endpoints differ from the Table 1 mean ratios by about one point.

C Extended Analyses

Why sampled training starves the ordering signal (from §4.3(a)). With the MovieLens-1M arm's 128 negatives per query against its 3,416-item catalog, a given competitor receives gradient with probability ≈0.04 per query (under one of a user's current top-20 competitors updated per step), while full-softmax CE updates all 3,415 every step. Its empirical fingerprint is present: NDCG improves proportionally more than recall in both pairs, and the same pathology is well documented for ID-softmax models [16, 27]. Two caveats. First, these are matched-configuration comparisons; we do not claim no sampled recipe can do better; no single loss dominates across datasets and evaluation setups, as measured on sequential ID-softmax models [15]. Second, sampled cross-entropy with large corrected negative pools (thousands of negatives) approaches full-softmax quality for ID-softmax models [16, 47], while our arms use the mixed in-batch recipe standard for dual encoders (§3.4).

Encoder-axis control (cold-start). Giving DropoutNet and Heater the *same* frozen text embeddings RetrievalFormer uses does not close their gap; it lowers their cold recall (Heater 0.087 → 0.054, to the level of the training-free floor; DropoutNet 0.077 → 0.059) while nudging warm up (DropoutNet text 0.211 ± 0.002 warm, Heater text 0.251 ± 0.003 warm; 3 seeds). Both baselines run at their reference configurations, whose capacities were designed for low-dimensional metadata rather than 1,536-dim text embeddings, so this control bounds the reference recipes, not the architectures' ceiling. Within that bound the conclusion is consistent across three probes (baselines-with-our-features, the last-item floor on byte-identical features, and our tower): the advantage comes from how the dual-encoder architecture uses the content encoder.

What "losing warm" means is protocol-relative. The cold benchmark's warm slice is a matrix-completion task (rank a user's held-out warm ratings) which rewards per-pair memorization, the home game of MF. Running both model classes through one shared harness in both directions illustrates the task-dependence descriptively (the two protocols differ in split, candidate set, and checkpoints; this is not a controlled decomposition): on the cold benchmark's warm slice, MF scores 0.267 and our content tower 0.113; on leave-one-out *next-item* prediction, our model scores 0.350 while MF scores 0.125 (popularity floor 0.068; MF cell single-seed). Each class wins its matched task, and only the feature-based tower also wins cold. "RetrievalFormer loses warm" is therefore a statement about that protocol's task, not about warm ability in general. The fusion recipe is protocol-specific in the same way: fusing frozen MF onto our model on the next-item benchmark does not help (fused 0.3469 vs. 0.3500 alone), so the operating point of Table 3 is a property of matrix-completion-style serving.

Within-class ablations (shared embeddings; attention mode). Within the dual-encoder class, sharing embedding tables across towers matters on average but not at the optimum: marginalized over our MovieLens-1M tuning grid (3 seeds per configuration), separating the towers' embedding tables costs 3.4% Recall@20 and removing shared features entirely costs 6.9%; at the single best configuration the gain is within seed noise. Shared embeddings buy robustness (one fewer thing to tune) without driving accuracy; we report marginalized deltas because grid-pooled absolutes depend on grid composition. A second ablation closes the attention-mode axis: replacing prefix-LM attention with pure causal attention (the only change is bidirectionality within the three-token profile/readout prefix, as history attention is causal in both modes) costs −0.0098 and −0.0026 test Recall@20 on paired seeds (pair mean 0.3627 vs. the constant-LR cap-300 family's 0.3668), so the bidirectional prefix block is load-bearing at the ∼+0.004 margin and the submitted configuration keeps it.