Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Exact Incremental Updates for Continual Sequential Recommendation

Emin Talip Demirkiran

etd@eskisehir.edu.tr

Eskisehir Technical University

Research Article

Keywords: Sequential recommendation, Continual learning, Incremental learning, Closed-form models, Recommender systems, Temporal recommendation

Posted Date: September 21st, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11048276/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Exact Incremental Updates for Continual Sequential Recommendation

Emin Talip Demirkiran[1*]

[1*]Department of Computer Engineering, Eskişehir Technical University, Eskişehir, Türkiye.

Corresponding author(s). E-mail(s): etd@eskisehir.edu.tr;

Abstract

Continual sequential recommendation must absorb newly arriving interactions without repeatedly retraining on all historical data, while preserving previously learned preferences. Existing work has largely pursued this goal with gradient-trained neural architectures, leaving the behavior of closed-form temporal recommenders under a genuine multi-block continual protocol insufficiently characterized. This study adapts a temporal linear item-item model to a causally valid setting and compare three update strategies on MovieLens-1M: full re-solving, sufficient-statistics incremental updating, and a Woodbury inverse update. Under the retained-accuracy/learning-accuracy protocol introduced for continual sequential recommendation, the closed-form model does not match the source-reported accuracy of CSTRec; its H-mean@20 gap widens across later blocks. However, the sufficient-statistics update is numerically equivalent to full re-solving to approximately $\mathbf{10^{-12}}$ relative error and is substantially faster after the initial block. The Woodbury formulation exposes a sharp boundary condition: because every update block is much larger than the item catalog, its dense update matrix causes deterministic memory-allocation failure. A causal, past-only trend-normalization correction also improves head- and tail-item accuracy in ablation. The results establish sufficient-statistics updating as an exact and computationally useful maintenance strategy within this closed-form model class, while delineating the regimes in which dense low-rank inverse updates and simplified linear recommenders are not viable substitutes for specialized continual neural models.

Keywords: Sequential recommendation, Continual learning, Incremental learning, Closed-form models, Recommender systems, Temporal recommendation

1 Introduction

Sequential recommendation predicts a user's next interaction from an ordered history of prior behavior. The field has progressed from Markov-chain and recurrent formulations to convolutional, graph-based, self-attentive, self-supervised, generative, and state-space architectures [3–16, 18, 21–24, 27–49]. This progression has improved representation capacity, but it has also increased the computational and engineering cost of repeatedly adapting models when interaction streams evolve. In deployed systems, the question is therefore not only how accurately a model predicts the next item, but also how it can incorporate new blocks of behavior without unnecessarily recomputing information already summarized by earlier data.

Continual sequential recommendation makes that tension explicit. CSTRec [25] evaluates sequential recommenders over multiple chronologically arriving blocks and separates retained accuracy on past behavior from learning accuracy on newly observed behavior. By contrast, TALE [20]

is a non-iterative temporal linear item-item model whose parameters are obtained from a closed-form ridge solution. Its algebra suggests that cumulative sufficient statistics can be updated exactly as new rows arrive, potentially avoiding repeated reconstruction of historical design matrices. Yet TALE was developed for a static setting, and its original trend-aware normalization uses information from both sides of a temporal window. That formulation is not causally valid when future blocks are unavailable at update time. Moreover, the practical value of closed-form incremental updates under the retention/acquisition trade-off used in continual sequential recommendation has not been established.

This study addresses that gap by asking what is gained and what is lost when a temporal closed-form sequential recommender is made genuinely incremental and evaluated under a multi-block continual protocol. Three implementations of the same model class are examined: a full cumulative re-solve, an exact sufficient-statistics update, and a Woodbury inverse update intended to exploit low-rank arrivals. The non-causal trend normalization is additionally replaced with a past-only variant and evaluated directly. The design separates four issues that are often conflated: absolute recommendation quality, numerical equivalence to full re-solving, computational cost, and the structural conditions under which a particular matrix-update identity is feasible.

The study distinguishes computational exactness from absolute ranking quality. On MovieLens-1M, the closed-form model trails the source-reported CSTRec results by a substantial and widening margin. At the same time, the sufficient-statistics formulation reproduces the full re-solve essentially to floating-point precision and reduces post-initial-block fitting time markedly. The attempted Woodbury formulation fails for a principled reason rather than an implementation accident: each update contains 26.6 to 172.3 times as many rows as there are catalog items, forcing construction of prohibitively large dense matrices. Finally, causal trend normalization produces large gains over disabling trend normalization in a controlled head/tail ablation.

The study makes three contributions. First, it provides an exact incremental formulation of a temporal closed-form sequential recommender and validates its numerical equivalence to full cumulative re-solving. Second, it characterizes an important boundary condition for dense Woodbury updates in recommender workloads, showing empirically that update rank must be treated as a first-class computational variable rather than assumed to be small. Third, it introduces and evaluates a causally valid, past-only trend-normalization correction required to transfer TALE-style temporal debiasing to a continual setting. Together, these findings clarify the realistic role of closed-form methods: they can offer transparent and efficient incremental maintenance within their model class, but they should not be presumed competitive with specialized neural continual models in absolute ranking quality.

2 Related Work

2.1 Sequential recommendation and representation learning

Sequential and session-based recommendation have evolved through several modeling paradigms. Early approaches captured local transitions with personalized Markov chains and recurrent networks, including FPMC [3] and GRU4Rec [4], followed by attention- and convolution-based models such as NARM [5], Caser [6], and SASRec [7]. Graph formulations including SR-GNN [8], GCE-GNN [28], and information-preserving graph propagation [31] model item-transition structure explicitly, while RepeatNet [34] and improved top-k recurrent objectives [32] address repeat behavior and ranking losses. Broad surveys of session-based, deep, and sequential recommendation [1, 2, 27] document the shift from compact transition models toward increasingly expressive sequence encoders. These works establish the representational foundations of the field, but they generally evaluate models in a static train/test regime rather than under repeated chronological updates.

A second line of work improves representation learning through self-supervision, contrastive objectives, and pre-training. S3-Rec [11], CL4SRec [12], DuoRec [13], DCRec [15], CoSeRec [29], and ICLRec [30] use auxiliary or contrastive signals to make sequence representations more

robust to sparsity, augmentation noise, representation degeneration, or latent intent. UniSRec [33] pursues transferable sequence representations, while MISSRec [35] extends pre-training to multi-modal interest-aware representations. Critical benchmarking work has also questioned whether contrastive learning itself is always necessary [16]. These methods are important context because they show how modern sequential recommendation improves accuracy through learned representations; however, their update process remains gradient-based and does not provide the exact algebraic equivalence between incremental and from-scratch solutions examined here.

2.2 Temporal, causal, and continual recommendation

Temporal structure, non-stationarity, and causality form the closest conceptual neighborhood to this study. TiSASRec [10] explicitly models time intervals, TALE [20] combines temporal weighting with a closed-form linear item-item solution, and TiSRec [23] segments sequences using temporal intervals. Interest drift and fatigue are modeled by IDURL [21] and FRec [17], while CauseRec [43] and CSRec [22] use counterfactual or causal perspectives to reduce spurious dependence on observed sequences. Continuous-time modeling has also been explored with temporal graph transformers [44] and, more recently, state-space models [45]. Most directly, CSTRec [25] formulates continual sequential recommendation over data streams and evaluates retention and acquisition across blocks. The unresolved intersection is whether a closed-form temporal model can be updated exactly and economically within that genuinely continual evaluation setting, and what causal adaptations are required when a static temporal normalization is moved online.

2.3 Generative and efficiency-oriented approaches

Generative and diffusion approaches provide another contrast in model complexity. DimeRec [24], sequential diffusion recommenders [37, 38], diffusion-based augmentation [39], and guided diffusion formulations [40] model uncertainty or generate preference-aligned representations rather than relying on a single deterministic linear map. These methods demonstrate the growing expressive range of sequential recommendation, but they also strengthen the motivation for understanding simpler baselines whose update behavior can be characterized analytically. The present study does not claim that a linear model should outperform such architectures; instead, it asks whether its closed-form structure yields a useful computational property under continual updates and where that property breaks down.

Efficiency-oriented work has recently expanded beyond conventional Transformers. FMLP-Rec [14] removes attention in favor of frequency-domain filtering; transformer scaling studies [19] examine capacity and efficiency trade-offs; selective state-space and Mamba-based recommenders investigate linear-time alternatives, including lifelong SSMs [36], M2Rec [42], SIGMA [46], MaTrRec [48], and Mamba4Rec [49]. Evaluation methodology is itself a source of uncertainty: sampling-based top-k evaluation can alter interpretations [47], and a recent graph-heuristic audit shows that some sequential benchmarks may contain shortcut structure [26]. Uniform Sequence Better [41] further highlights the role of temporal augmentation. This broader literature makes two points relevant to the design: computational efficiency must be measured rather than inferred from asymptotic labels, and benchmark-specific success should not be generalized beyond the tested regime. Against that background, the contribution examined here is the exact sufficient-statistics maintenance of a temporal linear recommender, its causal normalization for online use, and an empirical characterization of the update-rank boundary that defeats a naive dense Woodbury implementation.

3 Research Questions and Evaluation Criteria

Three primary research questions guide the evaluation. **RQ1** asks whether a closed-form temporal linear recommender is competitive with a specialized continual sequential architecture under the same retained-accuracy/learning-accuracy protocol. **RQ2** asks whether cumulative sufficient statistics reproduce the full cumulative re-solve within numerical precision while reducing update

cost. **RQ3** asks under which update-rank conditions a Woodbury inverse update is computationally feasible, with particular attention to the ratio between the number of newly arriving rows, r_t, and the item dimension, n. A separate mechanism analysis examines whether replacing future-aware trend normalization with a causal past-only formulation preserves useful debiasing behavior.

Evaluation criteria were specified before the reported experiments. Numerical equivalence was defined as a relative Frobenius error below 10^{-6} between the incremental and full solutions. Because a Woodbury update is expected to be advantageous only in the low-rank regime $r_t \ll n$, the ratio r_t/n was recorded for every block rather than assumed to be small. Absolute recommendation quality was compared with source-reported CSTRec and SASRec values under the same block protocol. The proposed closed-form fits are deterministic, whereas the external comparator values are published point estimates from a different execution environment; accordingly, the comparisons are reported as numerical differences rather than as inferential statistical tests.

4 Method

4.1 Design motivation

TALE's trend-aware popularity normalization uses both past and future N-day windows [20]. In a continual setting, future blocks are unavailable at update time, so the original formulation would introduce look-ahead information. The normalization is therefore modified to use only observations available before the prediction time. The effect of this causal, past-window-only correction is evaluated in Section 6.5.

4.2 Governing equations

Using TALE's regularized linear objective as the starting point [20], the causally constructed design and target matrices are denoted by $\tilde{S}_t$ and $\tilde{T}_t$, respectively.

The model minimizes the regularized least-squares objective

$$\hat{B}_t = \arg\min_B \|\tilde{T}_t - \tilde{S}_t B\|_F^2 + \lambda \|B\|_F^2,$$

with the corresponding closed-form solution

$$\begin{aligned} \hat{B}_t &= A_t^{-1} C_t \\ &= (\tilde{S}_t^\top \tilde{S}_t + \lambda I)^{-1} \tilde{S}_t^\top \tilde{T}_t. \end{aligned} \tag{1}$$

ARM-1 (full re-solve). Recompute A_t, C_t from all cumulative causally weighted data at each block and solve the system anew. The matrix inversion term is $O(n^3)$, while constructing the sufficient statistics also depends on the cumulative number of rows.

ARM-2 (sufficient-statistics incremental). $A_t = A_{t-1} + \Delta_t^\top \Delta_t$, $C_t = C_{t-1} + \Delta_t^\top \Delta \tilde{T}_t$; A_t is still inverted fresh each block ($O(n^3)$ in the item count n, not the cumulative row count), but the sufficient statistics are updated incrementally rather than recomputed from all historical rows.

ARM-3 (Woodbury-updated inverse incremental). With $U = \Delta_t^\top \in \mathbb{R}^{n \times r_t}$ (r_t = new rows in block t):

$$\begin{aligned} A_t^{-1} = A_{t-1}^{-1} \\ - A_{t-1}^{-1} U \big(I_{r_t} + U^\top A_{t-1}^{-1} U\big)^{-1} \\ \times U^\top A_{t-1}^{-1}. \end{aligned} \tag{2}$$

Cost $O(n^2 r_t + r_t^3)$ — cheaper than ARM-1/ARM-2's $O(n^3)$ only if $r_t \ll n$, an empirical condition that must be checked rather than assumed. Sections 6.3 and 6.4 evaluate this condition directly.

Numerical agreement. Agreement with the full re-solve is quantified by

$$\delta_t = \frac{\|\hat{B}_t^{\text{ARM-2}} - \hat{B}_t^{\text{ARM-1}}\|_F}{\|\hat{B}_t^{\text{ARM-1}}\|_F}.$$

A tolerance of $\delta_t < 10^{-6}$ is used for verification. Because ARM-1 and ARM-2 optimize the same regularized least-squares objective, any difference should arise only from floating-point arithmetic, provided that historical row weights remain unchanged after they enter the cumulative sufficient statistics. This condition is checked empirically through the block-wise agreement reported in Section 6.2.

4.3 Scope of the methodological contribution

The contribution does not lie in the original TALE objective and weighting scheme [20], the CSTRec

RA/LA/H-mean protocol [25], the Woodbury identity itself, or incremental linear-model updating in isolation. It lies in their continual-setting integration: (i) a causal past-window-only trend-normalization correction, (ii) an exact sufficient-statistics update evaluated under a genuine multi-block RA/LA/H-mean protocol, and (iii) a direct comparison between sufficient-statistics and Woodbury-based updates under an explicit update-rank feasibility condition. This framing separates estimator exactness, ranking quality, and computational feasibility rather than conflating them into a single efficiency claim.

5 Experimental Setup

5.1 Dataset and preprocessing

MovieLens-1M was used as the sole experimental dataset. After 5-core filtering with the CSTRec-compatible preprocessing rule, the data contain 6,028 users, 2,873 items, and 807,281 interactions. The chronological stream is divided into five continual blocks: D0 contains 60% of the interactions and serves as the pretraining block, while D1–D4 contain 10% each, following the block convention reported for CSTRec [25]. Yelp, the only additional dataset reported by both TALE and CSTRec, was not used because its Dataset Terms of Use acceptance had not been completed before the study. Accordingly, all new evidence reported here is restricted to MovieLens-1M, and no cross-dataset generality is claimed.

5.2 Hyperparameter selection

Hyperparameters were selected exclusively on a held-out validation split from D0 using the search ranges reported for TALE [20]: $\lambda \in \{1, 5, 10, 50, 100, 500, 1000\}$, $\tau_{\text{time}} \in [2^{-10}, 2^{10}]$ days in powers of two, $c \in [0, 1]$ in increments of 0.1, and trend-window length $N \in \{7, 30, 90, 180, 360, 720\}$ days. The selected configuration was then held fixed for D1–D4 to avoid test-block tuning. The final values were $\lambda = 5$, $\tau_{\text{time}} = 84.375\,\text{s}$ (2^{-10} days), $c = 0.2$, a 360-day trend window, and $\gamma = 0.5$. The selected temporal-decay value lies at the lower boundary of the search range while validation NDCG@20 was still improving toward that boundary; the implications of this boundary effect are discussed in Section 8.

5.3 Comparators

CSTRec values reported in its source paper [25] and the corresponding source-reported SASRec baseline are used as external performance references. They are not treated as same-environment reruns. The primary mechanism comparison is internal: ARM-1 versus ARM-2 and ARM-3. All three arms use identical data representations and temporal normalization and differ only in how $\hat{B}_t$ is computed, thereby isolating the effect of the update strategy. The absence of an independent CSTRec rerun is discussed in Section 8.

5.4 Implementation and reproducibility

Experiments were executed on Windows/MINGW64 with Python 3.11.9, NumPy 2.4.6, and SciPy 1.17.1 using CPU-based NumPy/SciPy linear algebra. The numerical environment was held constant across the reported runs. ARM-1/2/3 contain no random initialization or stochastic sampling, and a repeated ARM-1 execution reproduced all nine reported RA/LA/H-mean metrics exactly across the five blocks. Seed-based averaging is therefore not applicable to the proposed deterministic arms. Any significance markers associated with CSTRec/SASRec originate from the source paper and were not recomputed in this study.

5.5 Protocol consistency

The data split, hyperparameter-selection procedure, update definitions, and evaluation criteria were fixed before the reported test-block results were interpreted. ARM-1 and ARM-2 completed all five blocks. ARM-3 failed reproducibly at every block because the required dense update matrix exceeded feasible memory. Numerical agreement with ARM-1 was verified before ARM-2 accuracy results were interpreted, while the published CSTRec/SASRec values were used only as external context.

6 Results

6.1 Main results

As shown in Table 1, D0 is zero across all reported metrics because it serves as the pretraining base

Table 1 Retained accuracy (RA), learning accuracy (LA), and H-mean at Hit@20 (H@20), MRR@20 (M@20), and NDCG@20 (N@20) for ARM-1 and ARM-2 across continual blocks. ARM-1 and ARM-2 are numerically identical; see Section 6.2.

	RA			LA			H-mean		
Block	H@20	M@20	N@20	H@20	M@20	N@20	H@20	M@20	N@20
D0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
D1	0.3847	0.1188	0.1770	0.3847	0.1188	0.1770	0.3847	0.1188	0.1770
D2	0.3830	0.1181	0.1760	0.3799	0.1167	0.1742	0.3814	0.1174	0.1751
D3	0.3700	0.1128	0.1690	0.2396	0.0724	0.1089	0.2908	0.0882	0.1324
D4	0.3559	0.1078	0.1620	0.1378	0.0408	0.0619	0.1987	0.0592	0.0896

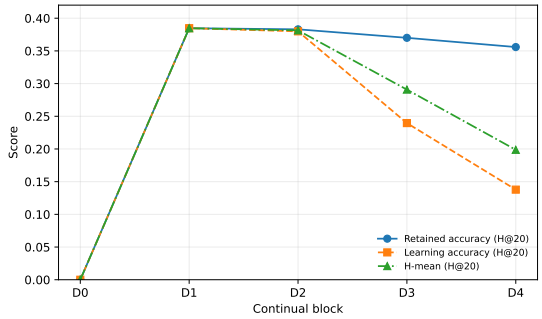


Fig. 1 Retained accuracy (RA), learning accuracy (LA), and H-mean@20 trajectories across continual blocks D0–D4.

block and therefore has no preceding/new-block pair for RA/LA evaluation. RA degrades gradually across subsequent blocks, whereas LA declines sharply from D2 to D4: the model's accuracy on the *current* block's new interactions deteriorates quickly as more blocks accumulate. H-mean, the harmonic mean of RA and LA used by CSTRec, is consequently constrained by the weaker component, producing the pronounced D3/D4 decline in the headline metric and the comparator gap discussed in Section 6.6.

ARM-3 produced no block-level accuracy results because the required dense allocation exceeded available memory before fitting could complete (Section 6.4).

Figure 1 plots the RA, LA, and H-mean@20 trajectories over D0–D4.

6.2 Numerical agreement

ARM-1 and ARM-2 are deterministic closed-form linear solves, so block-level agreement can be assessed directly without seed-based variance. Their relative Frobenius differences are D0 = 0.0, D1 = 3.921×10^{-13}, D2 = 7.210×10^{-13}, D3 = 9.909×10^{-13}, and D4 = 1.629×10^{-12}. All values are several orders of magnitude below the 10^{-6} verification threshold, demonstrating numerical equivalence within floating-point precision across all evaluated blocks.

6.3 Efficiency

Table 2 shows a substantial post-initialization reduction in fitting time for ARM-2 relative to ARM-1. Across D1–D4, ARM-2 requires 6.7–14.7 seconds per block compared with 51.5–125.0 seconds for ARM-1, while preserving the same solution within numerical precision. ARM-3's Woodbury-incremental approach is *not* cheaper than ARM-1/ARM-2 on this dataset — it never completes a fit at any block, because r_t/n is always in the unfavorable regime (26.6–172.3) that the method's own efficiency precondition ($r_t \ll n$) requires to hold in the opposite direction. Figure 2 summarizes the completed fitting-time trajectories. The favorable low-rank condition was therefore never instantiated on this dataset: the favorable low-rank regime is not evaluated here and remains a target for future study with genuinely small, frequent incremental blocks relative to a much larger catalog (Section 10). Timings were not collected in a dedicated, isolated benchmarking environment; therefore, the relative timing pattern and update-rank ratios are more informative than the absolute wall-clock values.

A direct efficiency comparison with CSTRec is not available: CSTRec was not rerun in the same environment, and CSTRec reports no training-cost/wall-clock figures to compare against. Accordingly, no claim of a computational advantage over CSTRec is made.

Table 2 Per-block fitting time and update-rank ratio. ARM-3 failed during dense matrix allocation at every block.

Block	r_t	n	r_t/n	ARM-1 (s)	ARM-2 (s)	ARM-3
D0	494,899	2,873	172.3	56.5	65.7	Failed (1.78 TiB attempted)
D1	76,693	2,873	26.7	51.5	8.2	Failed (43.8 GiB)
D2	76,815	2,873	26.7	77.5	10.9	Failed (44.0 GiB)
D3	76,336	2,873	26.6	69.2	6.7	Failed (43.4 GiB)
D4	76,510	2,873	26.6	125.0	14.7	Failed (43.6 GiB)

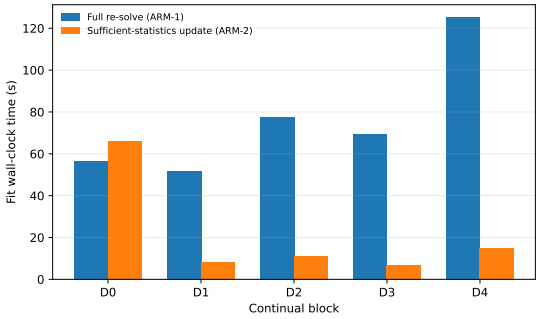


Fig. 2 Per-block fitting efficiency of the full re-solve (ARM-1) and sufficient-statistics incremental update (ARM-2). ARM-3 is omitted from completed-time trajectories because it fails during dense matrix allocation at every block.

6.4 Infeasibility of dense Woodbury updates in the tested regime

ARM-3 failed at all five blocks with a memory-allocation error while attempting to allocate the dense (r_t, r_t) matrix required by the Woodbury identity update.

The required dense allocations ranged from approximately 43.4–44.0 GiB in D1–D4 to 1.78 TiB in D0. The allocation failures were reproducible across blocks. The root cause is structural: the Woodbury formulation is attractive when $r_t \ll n$, whereas every block here has r_t 26.6–172.3 times larger than $n = 2{,}873$. A reduced-scale validation showed the same deterioration at a less extreme ratio ($r_t/n = 6.4$), where the update did not complete within 100 seconds.

The observed failure is specific to the block-size regime and dense implementation evaluated here. It should not be interpreted as a general limitation of Woodbury-style incremental updating, because settings with genuinely small r_t/n—for example, frequent micro-batches relative to a much larger catalog—are not evaluated in this study.

Table 3 Causal trend-normalization ablation on D0 using 500 evaluated instances per head/tail group.

Variant	Group	H@20	M@20	N@20
With trend norm	Head	0.8880	0.4516	0.5535
With trend norm	Tail	0.9640	0.6954	0.7603
Without trend norm	Head	0.5840	0.1802	0.2685
Without trend norm	Tail	0.4620	0.1005	0.1781

6.5 Ablation: causal trend normalization

A controlled D0-only ablation, restricted to the validation-stage setting to avoid test-block tuning, compares ARM-1 fit with vs. without the causal (past-window-only) trend normalization, 500 real evaluated instances per head/tail group (a bounded, non-exhaustive sample).

As shown in Table 3, N@20 improves by +106.15% (head, 0.2685 → 0.5535) and +327.0% (tail, 0.1781 → 0.7603) — a large, internally consistent effect supporting the conclusion that causal trend normalization improves both head and tail performance in this bounded mechanism analysis, with the larger relative gain observed for tail items. The ablation is interpreted as a deterministic mechanism check rather than a significance test; its bounded sample size is discussed in Section 8.

6.6 Comparison against CSTRec and SASRec

Table 4 shows that ARM-1/2 remains materially below the source-reported CSTRec values at D2–D4, with a widening gap across blocks. RQ1 is therefore not supported for the tested configuration. Because CSTRec was not rerun in the same execution environment, this comparison is interpreted as contextual evidence rather than a fully controlled head-to-head benchmark; Section 8 discusses this constraint.

Table 4 H-mean@20 compared with external reference values. No statistical test is applied to the deterministic-value-versus-cited-value comparison; the source paper reports comparator results only from D2 onward.

Block	ARM-1/2	SASRec[a]	CSTRec[a]	vs. SASRec	vs. CSTRec
D2	0.3814	0.6609	0.6696	-42.3%	-43.0%
D3	0.2908	0.5858	0.5940	-50.4%	-51.0%
D4	0.1987	0.4703	0.4784	-57.7%	-58.5%

[a]Source-reported external comparator values.

7 Discussion

The results separate the computational value of exact incremental maintenance from the representational capacity of the underlying recommender. The principal positive finding is the exactness of the sufficient-statistics update rather than an accuracy improvement over neural continual-learning models. ARM-2 reproduces the full cumulative re-solve to approximately 10^{-12} relative error across all blocks. This matters because it converts what could otherwise be treated as an engineering shortcut into an equivalent estimator: the historical data need not be reconstructed to obtain the same closed-form solution. The timing measurements support the corresponding systems benefit after initialization, although the non-isolated environment means the precise second counts should not be interpreted as a hardware benchmark.

The comparison with CSTRec places an equally important limit on that benefit. The H-mean@20 gap is already large at D2 and widens through D4, so computational simplicity does not compensate for the loss in ranking quality on this benchmark. The deterioration is driven especially by learning accuracy on later blocks, while retained accuracy degrades more gradually. This pattern is consistent with the interpretation that exact maintenance of a linear estimator does not by itself provide the adaptive representation capacity of a continual Transformer. It also argues against a common but weak inference in continual-learning work: immunity to optimization-based forgetting is not equivalent to competitiveness under distributional change. A model can be updated exactly and still be too restrictive to absorb new preference structure.

The widening gap should nevertheless not be attributed to model class alone with certainty. The temporal decay hyperparameter was selected at the lower boundary of the searched range, with the validation curve still improving toward that boundary. The search space was inherited from a non-continual setting rather than designed around repeated causal updates. Consequently, the experiment supports the conclusion that this tested configuration is not competitive with CSTRec; it does not establish that every closed-form temporal formulation must show the same gap. A continual-specific search over shorter decay scales is a concrete follow-up, but performing that search after seeing the test trajectory would have compromised the confirmatory comparison in the present study.

The Woodbury result provides a different kind of contribution: a sharp systems boundary condition. The identity is often presented as an efficient way to update an inverse after a low-rank change, but its usefulness depends on which dimension is actually small. Here the update rank equals the number of newly added rows and is far larger than the item dimension in every block. Constructing the required $r_t \times r_t$ dense matrix therefore turns an ostensibly incremental method into an impossible memory request. The failures are not evidence against Woodbury updates in general; they are evidence that block granularity and catalog size must be checked before selecting the formulation. In recommender systems with frequent micro-batches and very large catalogs, the same method may become attractive. A robust implementation should therefore switch formulations based on r_t/n or use chunked, iterative, or explicitly low-rank factorizations rather than allocating the dense update matrix unconditionally.

The causal trend-normalization ablation addresses a subtler methodological issue. A normalization that uses future observations is innocuous in a retrospective static experiment but leaks information in an online update protocol. Replacing that component with a past-only window did not merely preserve functionality; relative to disabling trend normalization, it improved both head and tail performance, with the larger gain on tail items. The ablation is bounded to a fixed sample and one dataset, so the

exact percentages should not be generalized. Still, the direction of the effect supports the broader design principle that temporal recommender components must be checked for causal validity before they are reused in continual evaluation.

Taken together, the findings suggest a practical division of labor. Exact sufficient-statistics updating is attractive when transparency, CPU execution, deterministic reproducibility, and fast maintenance of a linear model are priorities. Specialized continual neural architectures remain preferable when absolute ranking quality is the dominant objective and their training cost is acceptable. The contribution of the closed-form approach is therefore not that it replaces CSTRec, but that it supplies an analytically transparent baseline, an exact incremental maintenance rule, and a clearly measurable regime in which a seemingly efficient inverse update becomes structurally inappropriate.

8 Limitations and Threats to Validity

The findings should be interpreted in light of five main limitations. First, all new experimental evidence is restricted to MovieLens-1M. The conclusions about absolute accuracy, block geometry, and Woodbury failure therefore require replication on datasets with different catalog sizes and update granularities. Second, the CSTRec and SASRec comparator values are taken from the CSTRec source paper rather than produced by an independent rerun in the same hardware and software environment. Protocol fields and reported metrics were checked for compatibility, but source-reported comparisons cannot control every implementation detail; they should be interpreted as contextual performance references rather than a perfectly controlled head-to-head benchmark.

Third, the temporal hyperparameter search has a boundary effect: τ_{time} was selected at the smallest value in the searched range while the objective remained monotonic toward that boundary. This weakens causal attribution for the widening later-block accuracy gap and motivates a new, prospectively defined continual-specific search. Fourth, wall-clock measurements were collected on a non-isolated machine. The large post-initialization difference between full and sufficient-statistics updates is informative, but exact timings should not be treated as portable performance constants. Fifth, the causal-normalization ablation used 500 evaluated instances per head/tail group and is therefore a bounded mechanism check rather than a population-wide estimate.

Finally, the literature synthesis was designed to position the experiment rather than to constitute a systematic review and was not conducted under a PRISMA-style exhaustive retrieval protocol. Novelty claims are therefore limited to the specific methodological combination evaluated here rather than an unrestricted claim of priority across all incremental sequential-recommendation research.

9 Ethical Considerations

MovieLens-1M is a public benchmark distributed by GroupLens and uses pseudonymous identifiers. The reported study is an offline analysis of existing benchmark data and involves no participant recruitment, intervention, or new personal-data collection.

10 Practical Implications and Future Work

Future experiments should deliberately vary the geometry of incremental updates. The most direct test is to construct frequent micro-batches for which $r_t/n \ll 1$ and compare dense Woodbury, sufficient-statistics inversion, and iterative or chunked alternatives under identical accuracy and memory constraints. A catalog-aware switching rule can then be estimated rather than chosen heuristically. A second priority is a prospectively specified temporal hyperparameter search tailored to continual updates, including values below the current lower boundary. Third, CSTRec should be rerun under the same preprocessing and evaluation procedure so that model accuracy, fitting time, and resource use can be compared directly. Replication on a second shared dataset would then reveal whether the observed learning-accuracy decay and update-rank regime are specific to MovieLens-1M.

11 Conclusion

This study examined whether the algebraic simplicity of a closed-form temporal recommender translates into a useful continual-update mechanism. The answer is qualified. The sufficient-statistics formulation is an exact incremental replacement for full cumulative re-solving within numerical precision and offers a substantial post-initialization computational advantage within the same model class. A dense Woodbury update, however, becomes structurally infeasible when the arriving block is large relative to the item dimension, producing deterministic memory failure throughout the tested MovieLens-1M protocol. The causal past-only trend-normalization correction also improves both head- and tail-item accuracy relative to removing trend normalization.

These strengths do not erase the model's accuracy limitation: the tested closed-form formulation remains substantially below the source-reported CSTRec results, with a widening gap on later blocks. The resulting evidence therefore distinguishes computational exactness from predictive competitiveness. Closed-form incremental recommendation provides exact, transparent, CPU-friendly maintenance and a strong baseline for studying update mechanics, but it is not a substitute for specialized continual neural models when ranking quality is paramount. Future work should test genuinely low-rank update regimes, retune temporal decay prospectively for continual data, and replicate the comparison under a common execution environment and additional datasets.

Declarations

Funding. No funding was received for conducting this study.

Competing interests. The author has no relevant financial or non-financial interests to disclose.

Author contributions. Emin Talip Demirkiran is the sole author and was responsible for the study conception and design, methodology, implementation, analysis, interpretation of results, manuscript preparation, critical revision, and approval of the final manuscript.

Data availability. The MovieLens-1M dataset used in this study is publicly available from GroupLens at https://files.grouplens.org/datasets/movielens/ml-1m.zip. No proprietary dataset was used in the reported experiments.

Corresponding author. Emin Talip Demirkiran, Department of Computer Engineering, Eskişehir Technical University, Türkiye; e-mail: etd@eskisehir.edu.tr; phone: +90 554 612 3468; ORCID: https://orcid.org/0000-0001-6063-8178.

Supplementary Information

Additional metric tables, implementation and reproducibility details, and expanded efficiency and ablation results are provided in Online Resource 1.

References

[1] Pan, L.-W., Pan, W.-K., Wei, M.-Y., Yin, H.-Z., Ming, Z. (2026). A survey on sequential recommendation. Frontiers of Computer Science, 20, 2003606. https://doi.org/10.1007/s11704-025-41329-w.

[2] Wang, S., Cao, L., Wang, Y., Sheng, Q. Z., Orgun, M. A., Lian, D. (2021). A survey on session-based recommender systems. ACM Computing Surveys, 54(7), Article 154, 1–38. https://doi.org/10.1145/3465401.

[3] Rendle, S., Freudenthaler, C., Schmidt-Thieme, L. (2010). Factorizing Personalized Markov Chains for Next-Basket Recommendation. WWW 2010. https://doi.org/10.1145/1772690.1772773.

[4] Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D. (2016). Session-based Recommendations with Recurrent Neural Networks. ICLR 2016 / arXiv:1511.06939.

[5] Li, J., Ren, P., Chen, Z., Ren, Z., Lian, T., Ma, J. (2017). Neural Attentive Session-based Recommendation. CIKM 2017. https://doi.org/10.1145/3132847.3132926.

[6] Tang, J., Wang, K. (2018). Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. WSDM 2018. https://doi.org/10.1145/3159652.3159656.

[7] Kang, W.-C., McAuley, J. (2018). Self-attentive sequential recommendation. In: 2018 IEEE International Conference on Data Mining (ICDM), 197–206. https://doi.org/10.1109/ICDM.2018.00035.

[8] Wu, S., Tang, Y., Zhu, Y., Wang, L., Xie, X., Tan, T. (2019). Session-based recommendation with graph neural networks. Proceedings of the AAAI Conference on Artificial Intelligence, 33(1), 346–353. https://doi.org/10.1609/aaai.v33i01.3301346.

[9] Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., Jiang, P. (2019). BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. CIKM 2019. https://doi.org/10.1145/3357384.3357895.

[10] Li, J., Wang, Y., McAuley, J. (2020). Time Interval Aware Self-Attention for Sequential Recommendation. WSDM 2020. https://doi.org/10.1145/3336191.3371786.

[11] Zhou, K., Wang, H., Zhao, W. X., Zhu, Y., Wang, S., Zhang, F., Wang, Z., Wen, J.-R. (2020). S3-Rec: Self-Supervised Learning for Sequential Recommendation with Mutual Information Maximization. CIKM 2020. https://doi.org/10.1145/3340531.3411954.

[12] Xie, X., Sun, F., Liu, Z., Wu, S., Gao, J., Zhang, J., Ding, B., Cui, B. (2022). Contrastive learning for sequential recommendation. In: 2022 IEEE 38th International Conference on Data Engineering (ICDE), 1259–1273. https://doi.org/10.1109/ICDE53745.2022.00099.

[13] Qiu, R., Huang, Z., Yin, H., Wang, Z. (2022). Contrastive learning for representation degeneration problem in sequential recommendation. In: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining (WSDM), 813–823. https://doi.org/10.1145/3488560.3498433.

[14] Zhou, K., Yu, H., Zhao, W. X., Wen, J.-R. (2022). Filter-enhanced MLP is All You Need for Sequential Recommendation. WWW 2022 / arXiv:2202.13556. https://doi.org/10.1145/3485447.3512111.

[15] Yang, Y., Huang, C., Xia, L., Huang, C., Luo, D., Lin, K. (2023). Debiased Contrastive Learning for Sequential Recommendation. WWW 2023. https://doi.org/10.1145/3543507.3583361.

[16] Zhou, P., Huang, Y.-L., Xie, Y., Gao, J., Wang, S., Kim, J. B., Kim, S. (2024). Is Contrastive Learning Necessary? A Study of Data Augmentation vs Contrastive Learning in Sequential Recommendation. WWW 2024 / arXiv:2403.11136. https://doi.org/10.1145/3589334.3645661.

[17] Li, N., Ban, X., Ling, C., Gao, C., Hu, L., Jiang, P., Gai, K., Li, Y., Liao, Q. (2024). Modeling User Fatigue for Sequential Recommendation. SIGIR 2024. https://doi.org/10.1145/3626772.3657802.

[18] Choi, M., Kim, H., Cho, H., Lee, J. (2024). Multi-intent-aware Session-based Recommendation. SIGIR 2024 / arXiv:2405.00986. https://doi.org/10.1145/3626772.3657928.

[19] Zivic, P., Vazquez, H., Sanchez, J. (2024). Scaling Sequential Recommendation Models with Transformers. SIGIR 2024. https://doi.org/10.1145/3626772.3657816.

[20] Park, S., Yoon, M., Choi, M., Lee, J. (2025). Temporal Linear Item-Item Model for Sequential Recommendation (TALE). WSDM 2025 / arXiv:2412.07382. https://doi.org/10.1145/3701551.3703554.

[21] Lin, X., Pan, W., Ming, Z. (2025). Towards interest drift-driven user representation learning in sequential recommendation. In: Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, 1541–1551. https://doi.org/10.1145/3726302.3730099.

[22] Liu, X., Yuan, J., Zhou, Y., Li, J., Huang, F., Ai, W. (2024). CSRec: Rethinking Sequential Recommendation from A Causal Perspective. arXiv:2409.05872. https://doi.org/10.48550/arXiv.2409.05872.

[23] Kim, M., Kang, W., Kim, G.-W., Song, C. H., Lee, S., Choi, S.-M. (2025). End-to-End Time Interval-wise Segmentation for Sequential Recommendation. RecSys 2025 (short paper). https://doi.org/10.1145/3705328.3759327.

[24] Li, W., Huang, R., Zhao, H., Liu, C., Zheng, K., Liu, Q., Mou, N., Zhou, G., Lian, D., Song, Y., Bao, W., Yu, E., Ou, W. (2025). DimeRec: A unified framework for enhanced sequential recommendation via generative diffusion models. In: Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining (WSDM), 726–734. https://doi.org/10.1145/3701551.3703555.

[25] Lee, G., Yoo, H., Hwang, J., Kang, S., Yu, H. (2026). Capturing User Interests from Data Streams for Continual Sequential Recommendation (CSTRec). WSDM 2026 / arXiv:2506.07466. https://doi.org/10.1145/3773966.3777977.

[26] Han, H., Ma, L., Wang, H., Li, B., Zha, D., Tan, C. H., Gao, H., Liu, X., Moyerman, S., Katariya, S., Liu, H., Tang, J. (2026). An Embarrassingly Simple Graph Heuristic Reveals Shortcut-Solvable Benchmarks for Sequential Recommendation. arXiv:2605.07125v1 (preprint). https://doi.org/10.48550/arXiv.2605.07125.

[27] Fang, H., Zhang, D., Shu, Y., Guo, G. (2020). Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations. ACM Transactions on Information Systems, 39(1), Article 10, 1–42. https://doi.org/10.1145/3426723.

[28] Wang, Z., Wei, W., Cong, G., Li, X.-L., Mao, X.-L., Qiu, M. (2020). Global Context Enhanced Graph Neural Networks for Session-based Recommendation. SIGIR 2020. https://doi.org/10.1145/3397271.3401142.

[29] Liu, Z., Chen, Y., Li, J., Yu, P. S., McAuley, J., Xiong, C. (2021). Contrastive Self-supervised Sequential Recommendation with Robust Augmentation. arXiv:2108.06479. https://doi.org/10.48550/arXiv.2108.06479.

[30] Chen, Y., Liu, Z., Li, J., McAuley, J., Xiong, C. (2022). Intent Contrastive Learning for Sequential Recommendation. WWW 2022. https://doi.org/10.1145/3485447.3512090.

[31] Chen, T., Wong, C. W. (2020). Handling Information Loss of Graph Neural Networks for Session-based Recommendation. KDD 2020. https://doi.org/10.1145/3394486.3403170.

[32] Hidasi, B., Karatzoglou, A. (2018). Recurrent Neural Networks with Top-k Gains for Session-based Recommendations. CIKM 2018. https://doi.org/10.1145/3269206.3271761.

[33] Hou, Y., Mu, S., Zhao, W. X., Li, Y., Ding, B., Wen, J.-R. (2022). Towards Universal Sequence Representation Learning for Recommender Systems. KDD 2022. https://doi.org/10.1145/3534678.3539381.

[34] Ren, P., Chen, Z., Li, J., Ren, Z., Ma, J., de Rijke, M. (2019). RepeatNet: A Repeat Aware Neural Recommendation Machine for Session-Based Recommendation. AAAI 2019. https://doi.org/10.1609/aaai.v33i01.33014806.

[35] Wang, J., Zeng, Z., Wang, Y., et al. (2023). MISSRec: Pre-training and Transferring Multi-modal Interest-aware Sequence Representation for Recommendation. ACM Multimedia 2023. https://doi.org/10.1145/3581783.3611967.

[36] Yang, J., Li, Y., Zhao, J., et al. (2024). Uncovering Selective State Space Model's Capabilities in Lifelong Sequential Recommendation. arXiv:2403.16371. https://doi.org/10.48550/arXiv.2403.16371.

[37] Du, H., Yuan, H., Huang, Z., Zhao, P., Zhou, X. (2023). Sequential Recommendation with Diffusion Models. arXiv:2304.04541. https://doi.org/10.48550/arXiv.2304.04541.

[38] Li, Z., Sun, A., Li, C. (2023). DiffuRec: A diffusion model for sequential recommendation. ACM Transactions on Information Systems, 42(3), Article 66, 1–28. https://doi.org/10.

1145/3631116.

[39] Liu, Q., Yan, F., Zhao, X., Du, Z., Guo, H., Tang, R., Tian, F. (2023). Diffusion Augmentation for Sequential Recommendation. CIKM 2023. https://doi.org/10.1145/3583780.3615134.

[40] Yang, Z., Wu, J., Wang, Z., Wang, X., Yuan, Y., He, X. (2023). Generate what you prefer: Reshaping sequential recommendation via guided diffusion. Advances in Neural Information Processing Systems, 36.

[41] Dang, Y., Yang, E., Guo, G., Jiang, L., Wang, X., Xu, X., Sun, Q., Liu, H. (2023). Uniform Sequence Better: Time Interval Aware Data Augmentation for Sequential Recommendation. AAAI 2023. https://doi.org/10.1609/aaai.v37i4.25540.

[42] Zhang, Q., Qu, L., Wen, H., Huang, D., Yiu, S.-M., Hung, N. Q. V., Yin, H. (2026). M2Rec: Multi-scale Mamba for efficient sequential recommendation. IEEE Transactions on Knowledge and Data Engineering, advance online publication. https://doi.org/10.1109/TKDE.2026.3700323.

[43] Zhang, S., Yao, D., Zhao, Z., Chua, T.-S., Wu, F. (2021). CauseRec: Counterfactual User Sequence Synthesis for Sequential Recommendation. SIGIR 2021. https://doi.org/10.1145/3404835.3462908.

[44] Fan, Z., Liu, Z., Zhang, J., Xiong, Y., Zheng, L., Yu, P. S. (2021). Continuous-Time Sequential Recommendation with Temporal Graph Collaborative Transformer. CIKM 2021. https://doi.org/10.1145/3459637.3482242.

[45] Xiao, W., Wang, H., Zhou, Q., Wang, Q. (2025). SS4Rec: Continuous-Time Sequential Recommendation with State Space Models. arXiv:2502.08132. https://doi.org/10.48550/arXiv.2502.08132.

[46] Liu, Z., Liu, Q., Wang, Y., et al. (2025). SIGMA: Selective Gated Mamba for Sequential Recommendation. AAAI 2025. https://doi.org/10.1609/aaai.v39i12.33336.

[47] Li, D., Jin, R., Gao, J., Liu, Z. (2020). On Sampling Top-K Recommendation Evaluation. KDD 2020. https://doi.org/10.1145/3394486.3403262.

[48] Zhang, S., Zhang, R., Yang, Z. (2024). MaTrRec: Uniting Mamba and Transformer for Sequential Recommendation. arXiv:2407.19239. https://doi.org/10.48550/arXiv.2407.19239.

[49] Liu, C., Lin, J., Wang, J., Liu, H., Caverlee, J. (2024). Mamba4Rec: Towards Efficient Sequential Recommendation with Selective State Space Models. arXiv:2403.03900. https://doi.org/10.48550/arXiv.2403.03900.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- JDSASupplementaryMaterial.pdf