Supplementary Information

Online Resource 1

Exact Incremental Updates for Continual Sequential Recommendation

Emin Talip Demirkiran

Department of Computer Engineering, Eskişehir Technical University, Türkiye

Corresponding author: etd@eskisehir.edu.tr | +90 554 612 3468

ORCID: https://orcid.org/0000-0001-6063-8178

Purpose. This supplementary document provides protocol details, complete block-level metrics, numerical-agreement diagnostics, expanded efficiency and memory-allocation analyses, the causal trend-normalization ablation, and reproducibility information supporting the main manuscript. It does not introduce additional test-set experiments beyond those reported in the manuscript. Derived quantities such as speedup, time reduction, absolute comparator gaps, and ablation gains are calculated directly from the reported values.

Contents

S1. Study and Protocol Summary

Objective. The study evaluates whether a temporal closed-form sequential recommender can be updated incrementally under a genuine multi-block continual protocol while preserving the full cumulative solution, and it identifies the update-rank regime in which a dense Woodbury inverse update becomes infeasible.

Table S1. Study design and protocol summary.

Component	Specification
Dataset	MovieLens-1M
Filtered data	6,028 users; 2,873 items; 807,281 interactions
Continual split	D0 = 60%; D1-D4 = 10% each, chronological
Primary arms	ARM-1 full cumulative re-solve; ARM-2 sufficient-statistics incremental; ARM-3 Woodbury inverse update
Primary quality protocol	Retained accuracy (RA), learning accuracy (LA), and harmonic mean (H-mean) at Hit@20, MRR@20, and NDCG@20
Numerical-equivalence	Relative Frobenius error $\delta_t < 10^{-6}$

Component	Specification
threshold	
External comparators	Source-reported CSTRec and SASRec values; not same-environment reruns
Ablation	Causal past-only trend normalization vs. no trend normalization on D0; 500 evaluated instances per head/tail group
Execution mode	CPU-based deterministic NumPy/SciPy linear algebra; no random initialization or sampling in ARM-1/2/3

S2. Data, Split, and Hyperparameter Details

Data source. MovieLens-1M was obtained from GroupLens (https://files.grouplens.org/datasets/movielens/ml-1m.zip).

Table S2. Hyperparameter search space and final configuration.

Parameter	Search space	Selected value
Regularization λ	{1, 5, 10, 50, 100, 500, 1000}	5
Temporal decay parameter	$[2^{-10}, 2^{10}]$ days, powers of 2	84.375 s (2^{-10} days)
c	[0, 1] in increments of 0.1	0.2
Trend window N	{7, 30, 90, 180, 360, 720} days	360 days
γ	Fixed model setting reported in manuscript	0.5

Note. *Hyperparameters were selected only on a held-out D0 validation split and then fixed for D1-D4. The selected temporal-decay value lies at the lower search boundary while validation NDCG@20 was still improving toward that boundary; this is treated as a limitation rather than retuned after observing test-block behavior.*

S3. Closed-Form Update Formulations and Verification Criterion

Base estimator. Using causally constructed design and target matrices $\tilde{S}_t$ and $\tilde{T}_t$, the model solves a regularized least-squares problem with $\hat{B}_t = (\tilde{S}_t^T\tilde{S}_t + \lambda I)^{-1}\tilde{S}_t^T\tilde{T}_t$.

ARM-1 (full cumulative re-solve). At each block, cumulative sufficient statistics are reconstructed from all causally weighted rows and the regularized system is solved anew.

ARM-2 (sufficient-statistics incremental). $A_t = A_{t-1} + \Delta_t^T\Delta_t$ and $C_t = C_{t-1} + \Delta_t^T\Delta\tilde{T}_t$. The updated A_t is solved each block, but historical rows are not reconstructed.

ARM-3 (Woodbury inverse update). The inverse is updated through the Woodbury identity with $U = \Delta_t^T$. Its nominal advantage depends on the newly arriving rank r_t being much smaller than the item dimension n.

Verification metric. Numerical agreement is measured by $\delta_t = ||\hat{B}_t(\text{ARM-2}) - \hat{B}_t(\text{ARM-1})||F / ||\hat{B}_t(\text{ARM-1})||F$. The prespecified verification threshold is $\delta_t < 10^{-6}$. Because ARM-1 and ARM-2 optimize the same closed-form objective, discrepancies should be limited to floating-point arithmetic when historical row weights do not change after entering the cumulative statistics.

S4. Complete Block-Level Recommendation Metrics

ARM-1 and ARM-2 produce the same recommendation metrics within the verified numerical precision. The complete RA, LA, and H-mean values are reproduced below for convenience.

Table S3. Complete retained-accuracy, learning-accuracy, and H-mean results for ARM-1/2.

Block	RA H@20	RA M@20	RA N@20	LA H@20	LA M@20	LA N@20	H-mean H@20	H-mean M@20	H-mean 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

Note. *D0 is zero by protocol definition because it is the pretraining base block and has no preceding/new-block pair for RA/LA evaluation.*

S5. Numerical-Equivalence Diagnostics

Table S4 reports the relative Frobenius difference between ARM-2 and ARM-1 at each block. Every value is several orders of magnitude smaller than the 10^{-6} verification threshold.

Table S4. Block-wise numerical agreement between ARM-2 and ARM-1.

Block	Relative Frobenius error δ_t	$\delta_t < 10^{-6}$
D0	0.000	Pass
D1	3.921×10^{-13}	Pass
D2	7.210×10^{-13}	Pass
D3	9.909×10^{-13}	Pass
D4	1.629×10^{-12}	Pass

Interpretation. The largest observed discrepancy is 1.629×10^{-12} at D4, which remains about six orders of magnitude below the verification threshold. Thus, the sufficient-statistics formulation is empirically equivalent to the full cumulative re-solve to floating-point precision in the tested protocol.

S6. Efficiency and Update-Rank Diagnostics

The main manuscript reports per-block fitting times for ARM-1 and ARM-2 and the update-rank ratio r_t/n. Table S5 adds directly derived speedup and wall-clock reduction values to make the post-initialization efficiency pattern explicit.

Table S5. Per-block fitting time, update-rank ratio, and derived ARM-2 efficiency relative to ARM-1.

Block	r_t	n	r_t/n	ARM-1 (s)	ARM-2 (s)	Speedup	Time reduction
D0	494,899	2,873	172.3	56.5	65.7	0.86×	-16.3%
D1	76,693	2,873	26.7	51.5	8.2	6.28×	84.1%
D2	76,815	2,873	26.7	77.5	10.9	7.11×	85.9%
D3	76,336	2,873	26.6	69.2	6.7	10.33×	90.3%
D4	76,510	2,873	26.6	125.0	14.7	8.50×	88.2%

Note. *Speedup = ARM-1 time / ARM-2 time. Time reduction = 100 × (1 - ARM-2 time / ARM-1 time). D0 is an initialization case in which ARM-2 is slower; the efficiency advantage appears after initialization (D1-D4). Timings were not collected in an isolated benchmarking environment, so relative patterns are more informative than absolute seconds.*

S7. Dense Woodbury Memory-Allocation Analysis

ARM-3 requires a dense $r_t \times r_t$ intermediate matrix in the evaluated formulation. Because r_t is much larger than n in every block, this matrix dominates memory and prevents the update from completing.

Table S6. ARM-3 update-rank geometry and reported dense-allocation requests.

Block	r_t	n	r_t/n	Reported allocation	Outcome
D0	494,899	2,873	172.3	1.78 TiB	Failed at allocation
D1	76,693	2,873	26.7	43.8 GiB	Failed at allocation
D2	76,815	2,873	26.7	44.0 GiB	Failed at allocation
D3	76,336	2,873	26.6	43.4 GiB	Failed at allocation
D4	76,510	2,873	26.6	43.6 GiB	Failed at allocation

Boundary condition. The Woodbury formulation is attractive when the update rank is genuinely small relative to the item dimension. The tested MovieLens-1M block geometry is the opposite regime: r_t/n ranges from 26.6 to 172.3. A reduced-scale validation at $r_t/n = 6.4$ also did not complete within 100 seconds. These observations characterize the evaluated dense implementation and should not be generalized to micro-batch settings in which r_t/n is well below 1.

S8. Causal Trend-Normalization Ablation

The causal trend-normalization mechanism was evaluated on D0 using 500 real evaluated instances per head/tail group. Table S7 reproduces the raw metrics; Table S8 adds derived absolute and relative gains.

Table S7. D0 causal trend-normalization ablation.

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

Table S8. Derived gains from causal trend normalization relative to disabling trend normalization.

Group	Metric	Absolute gain	Relative gain
Head	H@20	+0.3040	+52.1%
Head	M@20	+0.2714	+150.6%
Head	N@20	+0.2850	+106.15%
Tail	H@20	+0.5020	+108.7%
Tail	M@20	+0.5949	+591.9%
Tail	N@20	+0.5822	+327.0%

Note. *Relative gains are deterministic arithmetic derived from the reported ablation values. Because the ablation is bounded to 500 instances per group and one dataset, it is a mechanism analysis rather than a population-wide significance claim.*

S9. External Comparator Context

CSTRec and SASRec values are source-reported external reference points rather than same-environment reruns. Table S9 reproduces H-mean@20 and adds absolute gaps alongside the relative differences reported in the main manuscript.

Table S9. H-mean@20 comparison with source-reported SASRec and CSTRec values.

Block	ARM-1/2	SASRec	Abs. gap vs SASRec	Rel. gap	CSTRec	Abs. gap vs CSTRec	Rel. gap
D2	0.3814	0.6609	-0.2795	-42.3%	0.6696	-0.2882	-43.0%
D3	0.2908	0.5858	-0.2950	-50.4%	0.5940	-0.3032	-51.0%
D4	0.1987	0.4703	-0.2716	-57.7%	0.4784	-0.2797	-58.5%

Note. *Absolute gap = ARM-1/2 minus comparator. Relative gaps are those reported in the manuscript. No inferential statistical test is applied because the proposed deterministic values and cited comparator values were not produced under one shared stochastic execution environment.*

S10. Reproducibility Checklist and Interpretation Boundaries

Table S10. Reproducibility and reporting checklist for the reported experiments.

Item	Detail	Status/interpretation
Software environment	Windows/MINGW64; Python 3.11.9; NumPy 2.4.6; SciPy 1.17.1	Reported
Hardware mode	CPU-based linear algebra; no GPU required for ARM-1/2/3	Reported
Random initialization	None in ARM-1/2/3	Not applicable
Stochastic sampling in fit path	None in ARM-1/2/3	Not applicable
Repeated deterministic check	Repeated ARM-1 run reproduced all nine RA/LA/H-mean metrics across five blocks	Verified
Hyperparameter tuning	D0 held-out validation only; fixed thereafter	Reported
Numerical-equivalence check	Block-wise ARM-2 vs ARM-1; threshold 10^{-6}	Verified
CSTRec rerun	Not performed in the same environment	Limitation
Wall-clock benchmark isolation	Not performed on a dedicated isolated benchmarking system	Limitation
Dataset coverage	MovieLens-1M only	Limitation
Ablation scope	500 instances per head/tail group on D0	Bounded mechanism check
Yelp	Not used because required terms acceptance was not completed	Excluded

Interpretation boundaries. The strongest claim supported by the study is exact incremental maintenance of the tested closed-form estimator through sufficient statistics, together with a clear update-rank feasibility boundary for the evaluated dense Woodbury implementation. The evidence does not support a claim that the closed-form model is competitive with CSTRec in absolute ranking quality, does not establish a same-environment efficiency advantage over CSTRec, and does not establish cross-dataset generality.

S11. Selected References

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

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

3. Kang, W.-C., McAuley, J. (2018). Self-Attentive Sequential Recommendation. 2018 IEEE International Conference on Data Mining (ICDM), 197-206. https://doi.org/10.1109/ICDM.2018.00035

End of Supplementary Information