PRECISION-WEIGHTED ATTENTION: RESTORING UNCERTAINTY TO THE TRANSFORMER FOR SEQUENTIAL RECOMMENDATION

Anonymous authors
Paper under double-blind review

ABSTRACT

The Transformer treats every token with uniform confidence: attention weights a position by how relevant it is to the query, never by how much evidence supports its representation. Sequential recommendation violates this assumption in every attention row, where an item observed a handful of times sits beside one observed thousands of times. We read the pre-norm Transformer layer as one cycle of a Bayesian filter: attention observes, the residual connection updates, and the feed-forward sublayer predicts. Under this reading exactly one quantity is missing from the standard layer, the precision of its state. Restoring it gives precision-weighted attention (PWA): each token carries a positive precision alongside its hidden state, and one observe–update–predict cycle per layer uses it as a bias on the attention logits toward better-supported positions, as the gain of the residual update, and as a quantity transported through the feed-forward sublayer by its Jacobian. The observation precision is closed-form, computed from the dispersion of the pooled values and their effective sample size, and PWA adds 132 parameters, four of which train, with no learned gate. On SASRec, BERT4Rec, and HSTU across six datasets, under one recipe fixed for every backbone and dataset, with 20 matched seeds per cell (720 runs), PWA improves mean NDCG@10 in 17 of 18 backbone–dataset pairs, 15 surviving Holm correction, with the largest gains on rare items and short histories. A matched-capacity control and component ablations locate the gain in the observe and update steps, and SASRec+PWA outperforms STOSA, the closest uncertainty-aware recommender, on every dataset. The filter reading is a structured approximation, not a calibrated posterior; we state each simplification and every negative cell. Precision as a first-class state modifies the Transformer itself, and its extension beyond recommendation is future work.

1 INTRODUCTION

Self-attention combines values with weights given by query–key similarity (Vaswani et al., 2017). The weights say how relevant each position is to the query, not how reliable it is: a key estimated from thousands of interactions and one estimated from three enter the same softmax on equal footing, and no state in a standard layer could carry such information from one layer to the next. Attention is a relevance model without an evidence model.

Sequential recommendation makes the cost concrete. A catalogue mixes items with long interaction histories and cold-start items observed a handful of times, so item representations differ by orders of magnitude in the evidence behind them (Dallmann et al., 2021; Damak et al., 2022), and user histories are similarly uneven. A standard recommender has no model-based way to discount a thinly observed item while still treating it as relevant; the usual remedies act on input construction or the loss and must be tuned per dataset.

We take the missing quantity from filtering. A Bayesian filter carries a belief about a latent state as a mean and a precision, and advances it by observing, updating, and predicting. A pre-norm Transformer layer performs three operations in the same order, attention, a residual update, and a feed-forward transition, and the correspondence identifies what the standard layer lacks: it observes without assessing how precise the observation is, updates at unit gain, and predicts without trans-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

porting confidence forward. Restoring precision gives precision-weighted attention (PWA): each token carries a positive feature-wise precision state initialized from item exposure; attention is biased toward better-supported positions and returns the precision of its pooled output; the residual update is scaled by a gain built from incoming and observed precision, in the manner of precision-weighted fusion (Kalman, 1960); and the feed-forward sublayer transports precision through its own Jacobian. Every step is one closed-form expression added to an otherwise unchanged layer.

Contributions.

1. A filter reading of the Transformer layer (Section 3) that identifies precision as the single missing state and prescribes, in closed form, the three places it enters: the attention logits, the residual gain, and the transition. The standard layer is the degenerate case of unit precision and unit gain.
2. PWA (Section 4), an instantiation whose observation precision is computed from the pooled values' dispersion and effective sample size (Kish, 1965) as the posterior mean of a Gamma pseudo-likelihood, with 132 added parameters, four trained, and no learned gate.
3. Evidence with attribution (Section 5): 720 matched-seed runs on three backbones and six datasets under one fixed configuration, Holm-corrected per-cell tests, cohort analyses along item exposure and history length, a comparison with STOSA (Fan et al., 2022), and a matched-capacity control with ablations that locate the gain in the observe and update steps.

The filtering vocabulary supplies a computational pattern, not an exact probabilistic model of a trained Transformer: we evaluate ranking quality, make no claim about calibration, and state every negative cell. Sequential recommendation is the setting throughout; the mechanism is a modification of the Transformer layer, and its extension to language modeling is future work (Section 6).

2 Related work

Sequential attention models. SASRec uses causal self-attention (Kang & McAuley, 2018), BERT4Rec a bidirectional masked-item formulation (Sun et al., 2019), and HSTU a gated transducer with non-softmax attention (Zhai et al., 2024). A well-trained SASRec with a full softmax loss is a strong baseline (Petrov & Macdonald, 2022; Klenitskiy & Vasilev, 2023; Petrov & Macdonald, 2023), which is why our comparisons are within-backbone contrasts under one full-vocabulary protocol rather than numbers copied from the original papers. **Noise and reliability in recommendation.** Rec-Denoiser learns differentiable masks over attention (Chen et al., 2022), AC-TSR calibrates attention weights (Zhou et al., 2023), and input-side denoising filters interactions before modeling (Wang et al., 2021); these learn which positions to suppress, whereas PWA computes a reliability for each pooled output in closed form inside an unchanged backbone. STOSA represents items by Gaussian embeddings with Wasserstein-based attention and ranking (Fan et al., 2022); it is the most direct uncertainty-aware reference and Section 5.4 compares against it. **Gates and uncertainty.** Learned sigmoid gates on the attention output (Qiu et al., 2025) and HSTU's SiLU gate modulate the residual with trainable capacity; the PWA gain has the same sigmoid form, $K = \sigma(\log(\lambda^{\mathrm{obs}}/\lambda))$, with a measured argument. Bayesian dropout and deep ensembles (Gal & Ghahramani, 2016; Lakshminarayanan et al., 2017) estimate output-level uncertainty by repeated prediction; PWA estimates neither, and ranking gains alone cannot establish calibration.

3 Framework: the Transformer layer as a Bayesian filter

3.1 The mapping, and what is missing

A Bayesian filter maintains a belief about a latent state as a mean h and a precision λ (an inverse variance, so larger means more confident) and advances it in three steps. Observe: form an estimate e from the current observation together with its precision λ^{obs}. Update: merge estimate and prior through the gain $K = \lambda^{\mathrm{obs}}/(\lambda + \lambda^{\mathrm{obs}})$, giving $h^{+} = h + K\,e$ and $\lambda^{+} = \lambda + \lambda^{\mathrm{obs}}$. Predict: evolve the state through a transition, deflating precision by the squared derivative of that transition. This is the extended Kalman filter in precision form (Kalman, 1960); its gain adapts on its own, passing a precise observation through ($K \to 1$) and suppressing an imprecise one ($K \to 0$) with no learned gate.

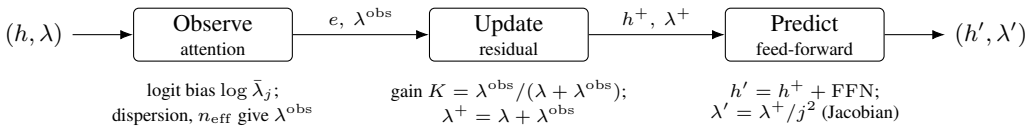


Figure 1: One PWA layer. The mean path is the standard pre-norm layer with a gain on its residual; the precision path uses λ as a log-precision bias on the attention logits ($\bar{\lambda}_j$, the mean incoming precision of key position j, Section 4.2), as the argument of the residual gain, and as a quantity transported through the transition by its diagonal Jacobian j (Section 4.5). n_{eff} is the effective sample size of an attention row (Section 4.3).

A pre-norm Transformer layer computes $h^+ = h + \mathrm{Attn}(\mathrm{LN}(h))$ and $h' = h^+ + \mathrm{FFN}(\mathrm{LN}(h^+))$, where LN is layer normalization, and each operation is one filter step. Attention is the observe step: it pools the values of surrounding positions into an estimate of what the current position should become, the role of a kernel-weighted estimator, and such an estimator carries a variance, the dispersion of what it pooled, which the standard layer discards. The residual connection is the update step at unit gain, $h^+ = h + e$ being $h^+ = h + Ke$ with $K \equiv 1$. The feed-forward sublayer is the predict step, and in a filter its derivative would carry precision forward. What is missing is therefore a single quantity: attention computes only the mean of its estimate, the residual applies it with no gain, and the transition moves no confidence. Precision is implicit and uniform, $\lambda_t = 1$ for every position t.

3.2 Restoring precision: the cycle

PWA adds exactly one state to the layer: a positive, feature-wise precision $\lambda_t \in \mathbb{R}^d_{>0}$ carried beside the hidden state $h_t \in \mathbb{R}^d$ of position t, where d is the model width. A layer maps $(h, \lambda) \mapsto (h', \lambda')$ through

$$\begin{aligned} &\text{(observe)} \quad (e_t, \lambda_t^{\mathrm{obs}}) = \mathrm{Observe}(h, \lambda)_t, \\ &\text{(update)} \quad K_t = \lambda_t^{\mathrm{obs}} \oslash (\lambda_t + \lambda_t^{\mathrm{obs}}), \quad h_t^+ = h_t + K_t \odot e_t, \quad \lambda_t^+ = \min\{C,\ \lambda_t + \lambda_t^{\mathrm{obs}}\}, \\ &\text{(predict)} \quad (h'_t, \lambda'_t) = \mathrm{Predict}(h_t^+, \lambda_t^+). \end{aligned} \tag{1}$$

Here $e_t \in \mathbb{R}^d$ is the pooled residual that attention returns for position t, $\lambda_t^{\mathrm{obs}} \in \mathbb{R}^d_{>0}$ is its estimated precision, $K_t \in (0,1)^d$ is the gain, $\odot$ and $\oslash$ are elementwise product and division, the minimum is taken coordinate-wise, and $C > 0$ is a cap that bounds accumulated precision ($C = 100$ throughout; Section 5.5 sweeps it). Observe runs the backbone's own attention, biased by the incoming precision, and estimates how precise its output is; Update is the ordinary residual connection given a gain; Predict is the backbone's transition sublayer, which advances the state and transports precision to the next layer's prior. Only λ is new, and every quantity is closed-form and elementwise (Figure 1).

Remark 1 (The standard layer is the degenerate case) *With uniform precision $\lambda \equiv 1$ the logit bias $\log \bar{\lambda}_j$ of Section 4.2 vanishes, and with $K \equiv 1$ the update is $h^+ = h + e$; the mean path is then exactly the standard pre-norm layer, whatever the precision path computes. PWA strictly generalizes the standard layer, and no single knob alone recovers it.*

3.3 What the reading is, and is not

The reading is a derivation device. It locates the missing quantity in all three steps at once, prescribes the form of the observation-precision estimator (the variance of a weighted local estimator, a solved problem in survey statistics), and predicts where gains should land: at rows with small effective sample size, at rare items, and at short histories. Section 5 tests that prediction.

The reading is not a claim that a trained layer computes a posterior. The derivation uses five simplifications, each stated where it enters: precision is feature-wise (no covariance); attention weights are treated as fixed when the variance is estimated; pooled values are treated as independent with a common mean and variance within a row; normalization and dropout Jacobians are omitted from transport; and the update adds $K \odot e$ rather than fusing two independent estimates of one state. Together these make PWA a trainable reliability-weighting rule with the structure of a filter, not

calibrated epistemic uncertainty, so we argue from the estimator the reading produces and from measurements, never from Bayesian optimality.

4 THE PWA ALGORITHM

This section specifies the four quantities the cycle leaves open, in order: the initial precision, the biased attention, the observation precision, and the transport. Indices: t and j run over sequence positions, f over the d features, ℓ over layers, and k over the hidden units of the feed-forward sublayer; head and batch indices are suppressed where they are not needed, and $\epsilon = 10^{-6}$ is a numerical guard.

4.1 INITIAL PRECISION FROM EXPOSURE

For an input sequence of item IDs $i_1, \ldots, i_T$, let $m_t \in \{0, 1\}$ mark non-padding positions and let $q_i \in [0, 1]$ be the frozen ordinal rank of item i by pre-test exposure count (0 for the rarest item, 1 for the most exposed; Appendix D states exactly which interactions are counted). Every feature of position t starts with

$$\lambda^0_{tf} = m_t \, \min\{C, \; R\, q_{i_t} + c\}, \qquad R = \mathrm{softplus}(a), \quad c = \mathrm{softplus}(a_0), \tag{2}$$

where f indexes features, and R and c are two global learned scalars, a range that controls how strongly exposure differentiates items and an offset that keeps every valid token's precision positive, starting at $R = 1$ and $c = \log 2$. Initialization is feature-isotropic and everything that follows is feature-specific; the state is recomputed for every sequence, so the only recursion is across layers.

4.2 OBSERVE: PRECISION-BIASED ATTENTION

In a head of width d_k, queries are projected from the normalized state, $q_t = W_Q \mathrm{LN}(h_t) + b_Q$, and keys and values from the raw state, $k_j = W_K h_j + b_K$ and $v_j = W_V h_j + b_V$, as in the backbone. Let $\bar{\lambda}_j = d_k^{-1} \sum_{f \in \text{head}} \lambda_{jf}$ be the incoming precision of position j averaged over the head's features, and $\alpha_{tj} = \mathrm{softmax}_j(s_{tj})$ the backbone's attention over the permitted keys (causal or bidirectional, padding excluded), with logits

$$s_{tj} = q_t^\top k_j / \sqrt{d_k} - \beta_\ell \, |t - j|, \qquad \beta_\ell = -\log \rho_\ell, \quad \rho_\ell = \mathrm{sigmoid}(r_\ell). \tag{3}$$

The second term is a relative-position bias with one learned distance logit r_ℓ per layer, shared across heads and initialized at $\rho_\ell = 0.99$; it expresses recency under causal masking and locality under bidirectional masking, alongside the backbone's absolute position embeddings. Precision then reweights the attention, which pools the values:

$$w_{tj} = \frac{\alpha_{tj} \, \bar{\lambda}_j}{\sum_r \alpha_{tr} \, \bar{\lambda}_r + \epsilon}, \qquad e_{tf} = \sum_j w_{tj} \, v_{jf}. \tag{4}$$

Up to attention dropout and the guard, Eq. 4 equals a single softmax over three additive terms, $w_{tj} = \mathrm{softmax}_j\big(q_t^\top k_j / \sqrt{d_k} + \log \bar{\lambda}_j - \beta_\ell |t - j|\big)$: content relevance, a reliability bias, and a positional preference (Appendix B.1). Precision weighting is therefore a data-dependent logit bias, one line in any softmax attention: a highly attended but thinly supported position is discounted, and a moderately attended but well-supported one can dominate. Heads are concatenated without an output projection.

4.3 OBSERVE: THE PRECISION OF THE POOLED RESIDUAL

Within one query and head, let n_t be the effective sample size of the row, in the sense of Kish (1965), and S_{tf} the dispersion of the pooled values around the pooled residual:

$$n_t = \Big(\textstyle\sum_j w_{tj}^2 \Big)^{-1}, \qquad S_{tf} = \max\Big\{ \textstyle\sum_j w_{tj} \, v_{jf}^2 - e_{tf}^2, \; 0 \Big\}, \qquad \lambda^{\mathrm{obs}}_{tf} = \min\Big\{ C, \; \tfrac{n_t \, (\nu + n_t - 1)}{\nu \sigma_0^2 + n_t \, S_{tf}} \Big\}, \tag{5}$$

with $\nu = 1$ and $\sigma_0^2 = 1/d_k$ fixed. n_t ranges from 1 (attention on one key) to the number of permitted keys (uniform attention). At fixed n_t, greater dispersion means lower precision; zero dispersion is

regularized rather than assigned infinite precision, and a row that attends to one key has bounded precision rather than a vanishing denominator. Both statistics are second moments of quantities attention already computes.

The form is not arbitrary. With fixed non-negative unit-mass weights and independent values of common mean and variance σ^2, $\mathrm{Var}(e) = \sigma^2/n$ and $\mathbb{E}[S] = \sigma^2(1-1/n)$, so nS carries the residual energy of $n-1$ degrees of freedom. Treating the corresponding Gaussian residual likelihood in $\tau = 1/\sigma^2$ as a pseudo-likelihood under a shape–rate $\mathrm{Gamma}(\nu/2, \nu\sigma_0^2/2)$ regularizer, the posterior mean of the pooled residual's precision is exactly the uncapped expression in Eq. 5 (Appendix B.2). $\sigma_0^2 = 1/d_k$ is the value-projection variance at initialization, and $\nu = 1$ is the smallest pseudo-count that keeps the estimate finite at $n_t = 1$, every sequence's first position under causal masking. Learned, correlated values do not satisfy these assumptions; the derivation motivates the estimator and does not certify a posterior.

4.4 Update: the precision-weighted residual

The gain is elementwise, and it scales an additive residual:

$$K_{tf} = \frac{\lambda_{tf}^{\mathrm{obs}}}{\lambda_{tf} + \lambda_{tf}^{\mathrm{obs}} + \epsilon}, \qquad h_t^+ = h_t + K_t \odot e_t, \qquad \lambda_t^+ = \min\{C,\ \lambda_t + \lambda_t^{\mathrm{obs}}\}. \tag{6}$$

The update adds $K \odot e$, not $K \odot (e - h)$: a gated residual, not a convex blend of two independent estimates of the state. Larger incoming precision resists change and larger observation precision admits more of the residual; key precision changes relative attention contributions through $\bar{\lambda}_j$, and query precision controls their residual impact through K_t, feature by feature.

4.5 Predict: transition and precision transport

The transition is the backbone's feed-forward sublayer, $z_t = \mathrm{LN}(h_t^+)$ and $h_t' = h_t^+ + \mathrm{FFN}(z_t)$ with $\mathrm{FFN}(z) = W_2\,\mathrm{ReLU}(W_1 z + b_1) + b_2$ (dropout omitted). First-order propagation through a map with derivative j multiplies variance by j^2, so precision is divided by it; we use the diagonal of the residual Jacobian,

$$\begin{aligned} j_{tf} &= 1 + \sum_k W_{2,fk}\,\mathbf{1}\big[(W_1 z_t + b_1)_k > 0\big]\,W_{1,kf}, \\ \lambda'_{tf} &= \min\Big\{C,\ \lambda_{tf}^+ / \max\{\mathrm{stopgrad}(j_{tf})^2,\ 0.01\}\Big\}, \end{aligned} \tag{7}$$

where $\mathbf{1}[\cdot]$ is the indicator function (the ReLU derivative), $\mathrm{stopgrad}$ detaches the transport factor so that it does not backpropagate into the FFN, and the floor 0.01 guards near-cancellation ($j \approx 0$). The transport is approximate: it omits the LayerNorm derivative, dropout, off-diagonal mixing, and covariance. Precision can decrease after the transition despite its accumulation in the update, which is the intended behavior: a coordinate the FFN has just reshaped is one the next layer should be less sure about. Padding precision is reset after each layer.

4.6 Instantiation, parameters, and cost

The cycle is stated for a generic attention operator: Observe needs pooling weights and a dispersion statistic, and Predict needs a transition, which may be the identity. SASRec and BERT4Rec (Kang & McAuley, 2018; Sun et al., 2019) instantiate it with softmax attention and a feed-forward transition as above; HSTU (Zhai et al., 2024) has no feed-forward sublayer and uses signed, unnormalized attention, so its transition is the identity, precision modulates its native gate, and the observation statistic is a regularized proxy. These are implementation details of the backbones, not part of the method; Appendix C gives them in full with the exact numerical order of every operation.

PWA adds 132 parameters to SASRec and BERT4Rec and 4 to HSTU, independent of catalogue size and under 0.1% of every evaluated model (Appendix E): the two global scalars of Eq. 2, one distance logit per layer, and 128 registered process-noise logits that are inert at the headline recipe (their ceiling is zero) and are counted for transparency. Only four SASRec additions receive gradient, so added capacity cannot explain the measured effect (Section 5.5 tests this directly). The cost is wall-clock: training time is $2.12\times$ the baseline on SASRec and $1.14\times$ on HSTU over 20 seeds

(Appendix E); the overhead sits on the encoder and shrinks as the output softmax grows with the catalogue.

5 EXPERIMENTS

5.1 PROTOCOL

Data and splits. MovieLens-1M and five Amazon categories: Musical Instruments, Sports and Outdoors, Toys and Games, Beauty, and Video Games. Interactions are chronologically ordered after iterative five-core filtering; the final item of each user is held out for testing and the penultimate for validation, and the test context includes the validation interaction. Statistics and per-dataset context limits (one position above the 99th percentile of evaluation-input length) are in Appendix D. **Training.** SASRec and HSTU are trained with full-vocabulary autoregressive cross-entropy. BERT4Rec uses full-vocabulary terminal-mask cross-entropy (the last retained training item is masked and predicted), which is cheaper than the original Cloze objective and does not reproduce published BERT4Rec effectiveness (Petrov & Macdonald, 2022); its rows are a matched within-backbone contrast, not numbers comparable to that literature. All within-backbone comparisons share objective, split, context limit, recipe, and seeds. **One recipe for every backbone and dataset.** The PWA recipe (Appendix D) was fixed once and applied unchanged to all three backbones and all six datasets. This is deliberate: the sweep asks whether PWA helps when dropped into an existing architecture without per-case tuning, so no setting was searched to favor any cell. The sensitivity analysis of Section 5.5 shows that this shared setting is not the best setting for any particular dataset or backbone, so the reported gains are a conservative estimate of what per-dataset and per-backbone tuning would deliver. **Evaluation.** Full-catalogue ranking of the held-out item, excluding padding and items in the retained context; sampled-negative evaluation need not preserve the ordering that full evaluation gives (Krichene & Rendle, 2020; Dallmann et al., 2021). HR@10 and NDCG@10 measure top-ten retrieval; MRR is untruncated. Each metric is averaged over test users within a run, then over 20 matched training seeds; two-sided paired Student tests are Holm-corrected within each table's family. Stars mark Holm-adjusted p ($^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$, $^{****}p < 0.0001$) and bold marks results surviving correction. Seed intervals describe training variability on fixed datasets, not uncertainty over future datasets. **Information boundary.** The exposure counts behind Eq. 2 include training interactions and validation-target identities but no test targets, so we call them pre-test exposure rather than training-only frequency; the prior is the only channel through which validation identities reach the model, and Section 5.5 includes an arm that disables it.

5.2 EXPERIMENT 1: COMPARISON WITH THE CORRESPONDING BACKBONES

Table 1 reports every backbone–dataset pair on all three metrics, one test per cell. Across the 18 pairs, mean NDCG@10 is higher with PWA in 17, and 15 of those differences remain significant after Holm correction over the 54-test family; HR@10 improves in 17 pairs (14 significant) and MRR in 17 (15 significant). The one exception is stated rather than averaged away: HSTU on Sports is lower on all three metrics without significance, and BERT4Rec on Instruments is higher on all three without significance. A non-significant difference is not evidence of equivalence. Effect sizes are single-digit relative on most cells and larger on ML-1M, where histories are long; the claim is consistency under one recipe shared across backbones and datasets, not the best number each cell can reach.

5.3 EXPERIMENT 2: RARE ITEMS AND SHORT HISTORIES

The reading predicts that the benefit concentrates where the evidence is thinnest. We distinguish two axes often conflated as cold start: item exposure (the pre-test catalogue frequency of the held-out target) and history length (the interactions available before the target). Each axis is split at its median, which puts the most observations behind every cell, with the same cohort assignment for both arms; metrics are user-averaged within every seed and the relative gain is the ratio of dataset-macro means over the 20 seeds (Appendix H). Table 2 reports SASRec with the cohort bounds and shares, because the shares explain the pattern.

Findings. On the exposure axis the rare half gains more than the popular half in 14 of 18 dataset–metric columns, and on the history axis the shorter half gains more than the longer half in 14 of

Table 1: Main sweep: paired mean HR@10, NDCG@10, and MRR (backbone vs. backbone+PWA) over 20 matched seeds per cell with raw two-sided paired-t p-values; bold marks the larger arm mean. Holm-adjusted p-values, standard deviations, and difference intervals for all 54 tests: Appendix F.

		HR@10			NDCG@10			MRR		
Dataset	Backbone	Base	PWA	p	Base	PWA	p	Base	PWA	p
Instruments	SASRec	0.1169	**0.1220**	1.7e-04	0.0649	**0.0680**	9.5e-06	0.0590	**0.0615**	3.7e-04
	BERT4Rec	0.0783	**0.0801**	1.6e-01	0.0427	**0.0435**	2.9e-01	0.0406	**0.0409**	6.3e-01
	HSTU	0.0946	**0.0985**	7.4e-03	0.0546	**0.0580**	5.6e-05	0.0505	**0.0538**	1.3e-05
Sports	SASRec	0.0491	**0.0529**	9.5e-12	0.0268	**0.0292**	1.1e-11	0.0251	**0.0271**	4.1e-11
	BERT4Rec	0.0260	**0.0280**	1.3e-07	0.0146	**0.0155**	8.7e-05	0.0143	**0.0149**	1.4e-03
	HSTU	**0.0437**	0.0433	2.1e-01	**0.0244**	0.0243	3.5e-01	**0.0229**	0.0228	3.8e-01
Toys	SASRec	0.0869	**0.0919**	4.8e-11	0.0508	**0.0542**	1.1e-10	0.0460	**0.0491**	2.3e-10
	BERT4Rec	0.0332	**0.0364**	1.5e-06	0.0197	**0.0214**	7.2e-05	0.0185	**0.0200**	8.2e-05
	HSTU	0.0677	**0.0702**	2.1e-04	0.0421	**0.0436**	1.3e-03	0.0388	**0.0402**	2.6e-03
Beauty	SASRec	0.0881	**0.0901**	5.6e-05	0.0501	**0.0513**	6.4e-05	0.0450	**0.0462**	1.8e-05
	BERT4Rec	0.0428	**0.0470**	8.3e-12	0.0240	**0.0261**	1.6e-10	0.0219	**0.0238**	4.7e-09
	HSTU	0.0713	**0.0727**	3.8e-02	0.0429	**0.0433**	3.1e-01	0.0393	**0.0396**	3.6e-01
Games	SASRec	0.1346	**0.1382**	5.3e-06	0.0722	**0.0749**	2.1e-07	0.0641	**0.0665**	3.0e-07
	BERT4Rec	0.0729	**0.0794**	3.1e-11	0.0384	**0.0417**	1.6e-10	0.0351	**0.0377**	1.4e-09
	HSTU	0.1221	**0.1247**	1.7e-04	0.0670	**0.0683**	7.7e-04	0.0601	**0.0611**	8.4e-04
ML-1M	SASRec	0.2341	**0.2524**	3.9e-14	0.1302	**0.1442**	7.0e-15	0.1133	**0.1258**	7.0e-15
	BERT4Rec	0.0665	**0.0943**	7.2e-15	0.0365	**0.0513**	2.1e-13	0.0343	**0.0472**	4.7e-13
	HSTU	0.2712	**0.2909**	3.1e-13	0.1566	**0.1703**	2.6e-14	0.1358	**0.1478**	1.4e-13

18. The exceptions are Instruments on all three metrics and ML-1M MRR on the first axis, and Toys on all three metrics and Beauty HR@10 on the second. The two axes are not independent, and the % pop. column measures the confound directly: on every dataset both user cohorts mostly target popular items (52–91% of their held-out items), and the short-history half does so slightly more than the long-history half on five of six datasets, by between 0.6 and 8 points. Conditioning on history length therefore mostly changes which popular items are predicted, the short-history cohort inherits the modest gains of the popular-item regime, and the history gradient is the weaker one; exposure carries the magnitude. These analyses are SASRec; in exploratory checks the cold-cohort advantage did not hold uniformly for the other backbones, so we claim it for SASRec only.

5.4 Experiment 3: comparison with STOSA

STOSA (Fan et al., 2022) is the closest uncertainty-aware sequential recommender, but it cannot be added on top of an existing backbone the way PWA can: it is a causal attention model of its own, with a mean and a covariance embedding per item, Wasserstein attention, and its own ranking objective. The natural comparison is between two causal models with uncertainty, STOSA and SASRec+PWA, under the same splits, context limits, and seed identifiers (Table 3). SASRec+PWA is higher on every dataset and metric, all 18 comparisons significant after Holm correction with 20 of 20 seeds in favor. The comparison uses the released STOSA implementation and is deliberately not capacity- or objective-matched; the remaining protocol differences (checkpoint selection, seen-item masking, MRR truncation on both sides) are listed in Appendix G. Two design differences plausibly contribute to the gap: STOSA doubles the per-item embedding parameters by adding a covariance embedding for every item, exactly where the data is thinnest, and its uncertainty is a learned per-item property rather than a quantity measured from the context of each query, so a rare item's reliability must be learned from the few interactions that mention it. The gap is a property of these two implementations, not evidence that distributional representations are inferior.

Table 2: Cohort analysis for SASRec (20 matched seeds). Cells are the relative change in each metric from adding PWA within one cohort. Top: item-exposure cohorts, Q1 the less exposed half of the catalogue and Q2 the more exposed half; Freq. is the cohort's range of pre-test exposure counts (zero-exposure targets form a separate cohort, Appendix H), % items the share of the catalogue, % test the share of scored test interactions. Bottom: history-length cohorts, Q1 the users with the shorter observed history; Len. is the cohort's range of observed history lengths after the context limit, % users the share of users, and % pop. the share of the cohort's held-out items that fall in the popular half of the catalogue (Appendix H). Stars and bold follow Section 5.1, Holm-corrected within each block's family; raw p-values in Appendix H.

Dataset	Cohort	Freq.	% items	% test	ΔHR@10	ΔNDCG@10	ΔMRR
Instruments	Q1 (rare)	1–7	60.1	45.6	-2.0%	-3.6%	$+0.9\%$
	Q2 (popular)	8–136	39.9	54.4	$\mathbf{+4.7\%}^{*}$	$\mathbf{+5.2\%}^{**}$	$\mathbf{+4.5\%}^{**}$
Sports	Q1 (rare)	1–8	55.0	33.9	$\mathbf{+36.3\%}^{**}$	$\mathbf{+43.0\%}^{**}$	$\mathbf{+25.6\%}^{***}$
	Q2 (popular)	9–963	45.0	66.1	$\mathbf{+7.1\%}^{****}$	$\mathbf{+8.2\%}^{****}$	$\mathbf{+7.4\%}^{****}$
Toys	Q1 (rare)	1–7	50.1	34.1	$\mathbf{+15.5\%}^{****}$	$\mathbf{+19.4\%}^{****}$	$\mathbf{+18.2\%}^{****}$
	Q2 (popular)	8–265	49.9	65.9	$\mathbf{+4.6\%}^{****}$	$\mathbf{+5.4\%}^{****}$	$\mathbf{+5.4\%}^{****}$
Beauty	Q1 (rare)	1–8	55.8	33.4	$\mathbf{+14.7\%}^{**}$	$\mathbf{+16.2\%}^{**}$	$\mathbf{+14.1\%}^{**}$
	Q2 (popular)	9–413	44.2	66.6	$\mathbf{+1.7\%}^{*}$	$\mathbf{+1.8\%}^{**}$	$\mathbf{+2.0\%}^{**}$
Games	Q1 (rare)	1–10	52.9	23.0	$+9.1\%$	$+8.1\%$	$+5.6\%$
	Q2 (popular)	11–618	47.1	77.0	$\mathbf{+2.5\%}^{***}$	$\mathbf{+3.6\%}^{****}$	$\mathbf{+3.8\%}^{****}$
ML-1M	Q1 (rare)	4–145	50.2	13.5	$\mathbf{+19.3\%}^{**}$	$\mathbf{+18.5\%}^{**}$	$\mathbf{+8.5\%}^{***}$
	Q2 (popular)	146–3403	49.8	86.5	$\mathbf{+7.7\%}^{****}$	$\mathbf{+10.7\%}^{****}$	$\mathbf{+11.1\%}^{****}$
Dataset	Cohort	Len.	% users	% pop.	ΔHR@10	ΔNDCG@10	ΔMRR
Instruments	Q1 (short)	4–5	59.8	56.3	$\mathbf{+5.0\%}^{**}$	$\mathbf{+5.1\%}^{***}$	$\mathbf{+4.6\%}^{**}$
	Q2 (long)	6–23	40.2	51.5	$+3.2\%$	$\mathbf{+4.0\%}^{*}$	$\mathbf{+3.6\%}^{*}$
Sports	Q1 (short)	4–5	51.4	66.5	$\mathbf{+8.3\%}^{****}$	$\mathbf{+10.1\%}^{****}$	$\mathbf{+9.2\%}^{****}$
	Q2 (long)	6–32	48.6	65.6	$\mathbf{+7.0\%}^{****}$	$\mathbf{+7.4\%}^{****}$	$\mathbf{+6.4\%}^{****}$
Toys	Q1 (short)	4–5	53.6	66.2	$\mathbf{+4.7\%}^{****}$	$\mathbf{+5.9\%}^{****}$	$\mathbf{+6.1\%}^{****}$
	Q2 (long)	6–40	46.4	65.6	$\mathbf{+7.1\%}^{****}$	$\mathbf{+7.8\%}^{****}$	$\mathbf{+7.3\%}^{****}$
Beauty	Q1 (short)	4–5	50.9	65.7	$\mathbf{+2.3\%}^{*}$	$\mathbf{+2.6\%}^{*}$	$\mathbf{+3.2\%}^{**}$
	Q2 (long)	6–43	49.1	67.3	$\mathbf{+2.3\%}^{**}$	$\mathbf{+2.3\%}^{**}$	$\mathbf{+2.2\%}^{*}$
Games	Q1 (short)	4–6	60.0	77.5	$\mathbf{+3.1\%}^{****}$	$\mathbf{+3.9\%}^{****}$	$\mathbf{+3.9\%}^{****}$
	Q2 (long)	7–51	40.0	76.2	$\mathbf{+2.0\%}^{*}$	$\mathbf{+3.3\%}^{***}$	$\mathbf{+3.8\%}^{****}$
ML-1M	Q1 (short)	17–95	50.3	90.5	$\mathbf{+8.1\%}^{****}$	$\mathbf{+12.3\%}^{****}$	$\mathbf{+13.5\%}^{****}$
	Q2 (long)	96–906	49.7	82.5	$\mathbf{+7.4\%}^{****}$	$\mathbf{+8.3\%}^{****}$	$\mathbf{+7.5\%}^{****}$

Table 3: STOSA vs. SASRec+PWA on 20 matched seeds: mean HR@10, NDCG@10, and MRR@40 (the released STOSA code logs a top-40-truncated MRR, applied to both sides), relative difference, and Holm-adjusted paired-t p over this table's 18 tests. Full statistics: Appendix G.

	HR@10			NDCG@10			MRR@40		
Dataset	STOSA	+PWA	Δ	STOSA	+PWA	Δ	STOSA	+PWA	Δ
Instruments	0.0795	**0.1220**	$+53.5\%^{****}$	0.0410	**0.0680**	$+65.7\%^{****}$	0.0343	**0.0574**	$+67.5\%^{****}$
Sports	0.0340	**0.0529**	$+55.7\%^{****}$	0.0186	**0.0292**	$+57.2\%^{****}$	0.0160	**0.0250**	$+56.0\%^{****}$
Toys	0.0705	**0.0919**	$+30.4\%^{****}$	0.0417	**0.0542**	$+30.1\%^{****}$	0.0358	**0.0468**	$+30.5\%^{****}$
Beauty	0.0631	**0.0901**	$+42.7\%^{****}$	0.0355	**0.0513**	$+44.6\%^{****}$	0.0304	**0.0438**	$+44.1\%^{****}$
Games	0.0894	**0.1382**	$+54.5\%^{****}$	0.0461	**0.0749**	$+62.3\%^{****}$	0.0387	**0.0630**	$+62.9\%^{****}$
ML-1M	0.1588	**0.2524**	$+59.0\%^{****}$	0.0857	**0.1442**	$+68.2\%^{****}$	0.0724	**0.1220**	$+68.5\%^{****}$

5.5 Experiment 4: attribution and sensitivity

Experiments 1–3 establish that PWA improves ranking and where, not which part of the cycle produces it. The most economical competing explanation is capacity: PWA adds parameters and changes the layer's carrier, the residual pathway that the attention output is added to and that feeds the feed-forward residual. PWA adds $K \odot e$ to the raw state h, whereas the backbone implementation adds e to the normalized state (Appendix C.3), so a baseline-versus-PWA comparison mixes the effect of precision with the effect of the carrier. The matched carrier keeps PWA's carrier, dropout placement, and parameter count and disables every precision computation, so the difference between it and full PWA is attributable to precision alone. Each other arm of Table 4 rebuilds the headline recipe on SASRec and disables one mechanism, over 5 seed-matched runs per cell; at this budget the entries are directional, not per-cell tests.

The matched-carrier control settles the capacity question: with the architecture, parameter budget, and residual pathway kept and every precision computation disabled, NDCG@10 drops by 3.41% on Beauty and 9.46% on ML-1M, among the largest movements in the table. Forcing $K \equiv 1$ costs

Table 4: Attribution on SASRec: change in NDCG@10 relative to full PWA when one mechanism is disabled (5 matched seeds per cell). The matched carrier keeps PWA's residual pathway (carrier), dropout placement, and parameter count and disables every precision computation, so it measures what the architecture change alone delivers.

Variant	Component removed	Beauty	ML-1M
Matched carrier (control)	every precision computation, carrier and parameters kept	-3.41%	-9.46%
No gain ($K \equiv 1$)	precision-weighted update	-1.85%	-3.44%
No attention precision	observation precision λ^{obs}	-1.07%	-0.25%
No item precision	exposure prior (Eq. 2)	$+0.61\%$	-0.28%
Log-frequency control	learned prior replaced by a frequency proxy	-2.14%	-9.84%
No position bias	distance term of Eq. 3	$+0.31\%$	-8.08%
No Jacobian transport	Eq. 7	$+0.62\%$	$+0.08\%$

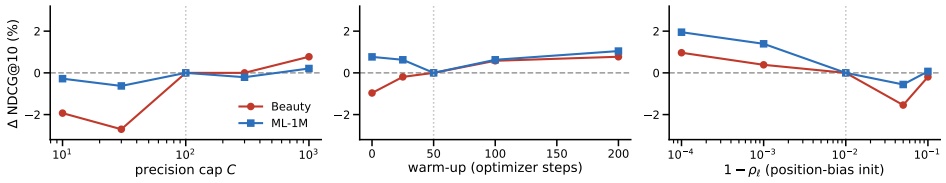


Figure 2: Sensitivity of NDCG@10 to the three settings of the headline recipe, each swept alone over 5 matched seeds on Beauty and ML-1M, as the relative change against the headline setting (dotted line, the published run; values in Appendix I). No swept setting differs significantly from the reference after Holm correction, and in every sweep some non-reference value edges past the published setting, so the recipe is not the argmax of this grid.

3.44% on ML-1M and 1.85% on Beauty, so the precision-weighted update is load-bearing on both datasets but not alone, since neither loss approaches the carrier's. Replacing the learned exposure prior by a log-frequency proxy costs 9.84% on ML-1M and 2.14% on Beauty: a prior that merely tracks popularity does not recover the gain. Removing the distance term is the largest single removal on ML-1M (8.08%), the dataset with long contexts, and is inert on Beauty. The individual precision estimates behave differently: on ML-1M, disabling attention precision, item precision, or Jacobian transport each moves the model by under a point, yet the carrier, which disables all of them at once, gives up 9.46%, so precision is encoded redundantly across sources and no single estimate is necessary while the ensemble is. For the filter reading this means that at depth two the observe and update steps carry the gain and the predict step's transport is not load-bearing, which licenses a cheaper approximation of the costliest component rather than showing transport inert.

Figure 2 asks how carefully the recipe had to be tuned: it sweeps the precision cap C over two orders of magnitude, the warm-up over 0–200 steps, and the position-bias initialization ρ_ℓ from 0.9 to 0.9999; no setting moves NDCG@10 by more than 2.73% and none of the 24 swept cells differs significantly from the reference. The sweep also makes the shared-recipe point concrete: on both datasets some non-reference value of every setting edges past the published one, so the headline numbers are what one general setting delivers, and per-dataset, per-backbone tuning is expected to improve on them.

6 Discussion, Limitations, and Future Directions

The evaluated configurations are reliability-aware ranking models, not calibrated posterior estimators (Section 3.3). The mechanism modifies the Transformer layer, not a recommender. In supervised fine-tuning of language models on noisy corpora, tokens differ in the reliability of their supervision and context rather than in exposure, so the exposure prior of Eq. 2 would give way to a signal-quality prior with the rest of the cycle unchanged; and the unit-gain update of Remark 1 suggests a hypothesis about attention sinks, which it can avoid only by attending to a small-norm value. Both are untested hypotheses.

USE OF AI-ASSISTED TOOLS

AI-assisted tools were used for code investigation and manuscript polishing. Scientific claims, experimental decisions, and the final submission remain the authors' responsibility.

ETHICS STATEMENT

The experiments use existing public recommendation benchmarks and introduce no new collection of personal data. Ranking gains do not establish fairness, privacy, causal user benefit, or deployment readiness. Frequency-based priors can reflect historical exposure disparities, which makes the cohort analysis of Section 5.3 important even when aggregate accuracy improves.

REPRODUCIBILITY STATEMENT

All numerical tables and cohort analyses are generated from per-seed result records or saved per-user ranks, and the analysis retains raw and adjusted p-values, pairing identifiers, cohort definitions, and source-file hashes. Appendices C and D give the exact numerical order of every operation, the recovered checkpoint configurations, the effective recipes, and the data boundaries. Code, configurations, and per-seed results are not attached to this submission; they will be released after the institutional review that our release process requires.

REFERENCES

Huiyuan Chen, Yusan Lin, Menghai Pan, Lan Wang, Chin-Chia Michael Yeh, Xiaoting Li, Yan Zheng, Fei Wang, and Hao Yang. Denoising self-attentive sequential recommendation. In *Proceedings of the 16th ACM Conference on Recommender Systems (RecSys '22)*, pp. 92–101, 2022. doi: 10.1145/3523227.3546788.

Alexander Dallmann, Daniel Zoller, and Andreas Hotho. A case study on sampling strategies for evaluating neural sequential item recommendation models. In *Proceedings of the 15th ACM Conference on Recommender Systems (RecSys '21)*, pp. 505–514, 2021. doi: 10.1145/3460231.3475943.

Khalil Damak, Sami Khenissi, and Olfa Nasraoui. Debiasing the cloze task in sequential recommendation with bidirectional transformers. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22)*, pp. 273–282, 2022. doi: 10.1145/3534678.3539430.

Ziwei Fan, Zhiwei Liu, Yu Wang, Alice Wang, Zahra Nazari, Lei Zheng, Hao Peng, and Philip S. Yu. Sequential recommendation via stochastic self-attention. In *Proceedings of the ACM Web Conference 2022 (WWW '22)*, pp. 2036–2047, 2022. doi: 10.1145/3485447.3512077.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1050–1059, 2016.

Rudolf E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 197–206, 2018. doi: 10.1109/ICDM.2018.00035.

Leslie Kish. *Survey Sampling*. Wiley, 1965.

Anton Klenitskiy and Alexey Vasilev. Turning dross into gold loss: is BERT4Rec really better than SASRec? In *Proceedings of the 17th ACM Conference on Recommender Systems (RecSys '23)*, pp. 1120–1125, 2023. doi: 10.1145/3604915.3610644.

Walid Krichene and Steffen Rendle. On sampled metrics for item recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '20)*, pp. 1748–1757, 2020. doi: 10.1145/3394486.3403226.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pp. 6402–6413, 2017.

Aleksandr Petrov and Craig Macdonald. A systematic review and replicability study of BERT4Rec for sequential recommendation. In *Proceedings of the 16th ACM Conference on Recommender Systems (RecSys '22)*, pp. 436–447, 2022. doi: 10.1145/3523227.3548487.

Aleksandr Petrov and Craig Macdonald. gSASRec: Reducing overconfidence in sequential recommendation trained with negative sampling. In *Proceedings of the 17th ACM Conference on Recommender Systems (RecSys '23)*, pp. 116–128, 2023. doi: 10.1145/3604915.3608783.

Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, Rui Men, Le Yu, Fei Huang, Suozhi Huang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Gated attention for large language models: Non-linearity, sparsity, and attention-sink-free. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*, 2025.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM '19)*, pp. 1441–1450, 2019. doi: 10.1145/3357384.3357895.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 2017.

Wenjie Wang, Fuli Feng, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. Denoising implicit feedback for recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining (WSDM '21)*, pp. 373–381, 2021. doi: 10.1145/3437963.3441800.

Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Jiayuan He, Yinghai Lu, and Yu Shi. Actions speak louder than words: Trillion-parameter sequential transducers for generative recommendations. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 58484–58509, 2024. URL https://proceedings.mlr.press/v235/zhai24a.html.

Peilin Zhou, Qichen Ye, Yueqi Xie, Jingqi Gao, Shoujin Wang, Jae Boum Kim, Chenyu You, and Sunghun Kim. Attention calibration for transformer-based sequential recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23)*, pp. 3595–3605, 2023. doi: 10.1145/3583780.3614785.

A NOTATION

Symbol	Where	Meaning
$h_t \in \mathbb{R}^d$, $\lambda_t \in \mathbb{R}^d_{>0}$	Eq. 1	Hidden state and feature-wise precision of position t; d is the model width
e_t, λ_t^{obs}	Eq. 1	Pooled residual returned by attention and its estimated precision
K_t	Eq. 6	Elementwise gain of the residual update
h_t^+, λ_t^+; h'_t, λ'_t	Eq. 1	State and precision after the update; after the transition (next layer's prior)
C	Eq. 1	Precision cap, $C = 100$
ϵ	Section 4	Numerical guard, 10^{-6}
m_t	Eq. 2	Validity mask, 1 on non-padding positions
q_i, R, c	Eq. 2	Ordinal exposure rank of item i in $[0, 1]$; learned global range and offset
q_t, k_j, v_j; d_k	Eq. 3	Query, key, value projections; head width
α_{tj}, w_{tj}	Eqs. 3, 4	Backbone attention weights; precision-reweighted (kriging) weights
$\bar{\lambda}_j$	Eq. 4	Incoming precision of position j averaged over a head's features
$\rho_\ell, \beta_\ell, r_\ell$	Eq. 3	Per-layer distance factor, its negative log, and its logit
n_t, S_{tf}	Eq. 5	Kish effective sample size of a row; dispersion of the pooled values
ν, σ_0^2	Eq. 5	Pseudo-count and scale of the Gamma regularizer, 1 and $1/d_k$
j_{tf}	Eq. 7	Diagonal of the residual Jacobian of the transition
W_Q, W_K, W_V; W_1, W_2, b_1, b_2	Sections 4.2, 4.5	Attention and feed-forward weights of the backbone
t, j; f; ℓ; k	Section 4	Indices over positions; features; layers; feed-forward hidden units
$\mathbf{1}[\cdot]$	Eq. 7	Indicator function
D_p	Appendix C	Inverted dropout with rate p (identity at evaluation)

B DERIVATIONS

B.1 THE LOG-PRECISION IDENTITY AND THE POSITION BIAS

Write the backbone logits as $s^c_{tj} = q_t^\top k_j/\sqrt{d_k}$. Since $\bar{\lambda}_j \exp(s^c_{tj}) = \exp(s^c_{tj} + \log \bar{\lambda}_j)$, the precision-reweighted weights of Eq. 4 equal a softmax over $s^c_{tj} + \log \bar{\lambda}_j$; multiplying every contribution by $\exp(-\beta_\ell|t - j|)$ adds the relative-position term. With content scores and precisions held fixed, increasing the distance by one multiplies a key's unnormalized contribution by $\exp(-\beta_\ell) = \rho_\ell$. Equal key precisions cancel under ideal softmax normalization, and $\rho_\ell = 1$ removes the position bias. These are identities about attention weights; they do not make whole models equivalent. The implemented order differs from the idealized identity in two ways that matter numerically: attention dropout is applied to α before precision reweighting, so the surviving weights are renormalized by the same denominator that carries the guard ϵ, and that denominator is additive rather than clamped. Both the residual e and its precision are computed from the same surviving weights. The guarded parameterization is $\beta_\ell = -\log \max\{\text{sigmoid}(r_\ell), 10^{-8}\}$ with r_ℓ initialized at $\log(0.99/0.01)$.

B.2 THE POSTERIOR-FORM OBSERVATION PRECISION

Consider one coordinate of one attention row, temporarily assuming fixed weights $w_j \geq 0$ with $\sum_j w_j = 1$ and independent values $v_j = \mu + \eta_j$ with zero-mean noise of common variance σ^2. Writing $e = \sum_j w_j v_j$ and $S = \sum_j w_j v_j^2 - e^2$, independence gives

$$\text{Var}(e) = \sigma^2 \sum_j w_j^2 = \frac{\sigma^2}{n}, \qquad n = \frac{1}{\sum_j w_j^2}, \qquad \mathbb{E}[S] = \sigma^2\Big(1 - \frac{1}{n}\Big), \tag{8}$$

so nS has the expected residual energy of $n-1$ homoscedastic degrees of freedom. With n equally weighted independent Gaussian samples, the residual likelihood after eliminating the unknown mean has the form below; for arbitrary weights and non-integer n we adopt the same effective-sample form as a pseudo-likelihood (moment matching alone does not establish its distribution):

$$L(\tau; S, n) \propto \tau^{(n-1)/2} \exp\Big(-\frac{nS\tau}{2}\Big), \qquad \tau = \sigma^{-2}. \tag{9}$$

A shape–rate $\mathrm{Gamma}(\nu/2, \nu\sigma_0^2/2)$ regularizer on τ is conjugate to it:

$$\tau \mid S, n \sim \mathrm{Gamma}\Big(\frac{\nu+n-1}{2}, \frac{\nu\sigma_0^2+nS}{2}\Big), \qquad n\,\mathbb{E}[\tau \mid S, n] = \frac{n(\nu+n-1)}{\nu\sigma_0^2+nS}. \tag{10}$$

The last expression is the posterior mean of the pooled residual's precision in this surrogate (the precision of the mean is $n\tau$), not the reciprocal of its posterior-mean variance. With $\nu = 1$ and $\sigma_0^2 = 1/d_k$, and the cap at C, this is Eq. 5. At $n = 1$ the estimate is $1/(\sigma_0^2 + S) \le d_k$; as n grows with S fixed it approaches n/S, the plug-in variance of the mean. The encoder does not satisfy the assumptions: attention weights depend on related hidden states, values are correlated, and the residual and its incoming state are not independent measurements. Numerical weight normalization also need not have unit mass. "Posterior" therefore identifies a regularized estimator motivated by this surrogate, not exact inference or calibrated predictive uncertainty. No such derivation is asserted for the signed HSTU weights (Appendix C.2).

B.3 The transport step in full

The diagonal j_{tf} of Eq. 7 is the exact diagonal of the dropout-free map $z \mapsto z + \mathrm{FFN}(z)$ evaluated at $z_t = \mathrm{LN}(h_t^+)$, but not of the implemented $h^+ \mapsto h^+ + \mathrm{FFN}(\mathrm{LN}(h^+))$: neither the LayerNorm derivative nor either FFN dropout mask is included, and off-diagonal contributions and covariances are omitted. The guarded transition as implemented is

$$\begin{aligned} d_{tf}^2 &= \max\{\mathrm{stopgrad}(j_{tf})^2, 0.01\}, & v_{tf}^F &= \frac{d_{tf}^2}{\max\{\lambda_{tf}^+, 10^{-6}\}}, \\ \lambda_{tf}^F &= \min\Big\{C, \frac{1}{\max\{v_{tf}^F, 10^{-6}\}}\Big\}, & \lambda'_{tf} &= \frac{m_t}{\max\{1/\max\{\lambda_{tf}^F, 10^{-4}\}, 10^{-8}\}}. \end{aligned} \tag{11}$$

For finite $\lambda^F \le C$ the last expression equals $m_t \max\{\lambda_{tf}^F, 10^{-4}\}$ in real arithmetic. No process noise is added at the headline recipe: the process-noise ceiling is zero, so the 128 registered process-noise logits per model are inert. Detachment prevents the transport factor from backpropagating to the FFN or its input, but the upstream precision still receives gradients through the division, and the FFN parameters train through the hidden-state path.

C Backbone instantiations and exact numerical order

All tensor products, divisions, clipping operations, and mask multiplications are elementwise unless a matrix product or sum is shown. There are two layers, two heads, $d = 64$, hence $d_k = 32$; D_p is inverted Bernoulli dropout during training and the identity at evaluation, and distinct calls use independent masks. In Algorithm 1, $\mathrm{SDPA}(q, k, v, M, p)$ is standard scaled dot-product attention under mask M with attention dropout p, $\mathrm{ConcatHeads}$ concatenates the per-head outputs along the feature axis, and LN_a, LN_f, and $\mathrm{LN}_{\mathrm{final}}$ are the backbone's attention, feed-forward, and output layer normalizations.

C.1 Softmax encoders (SASRec, terminal-masked BERT4Rec)

Exposure rank. Let N be the length of the item-count vector, including special-token slots. The frozen buffer is $q_i = \mathrm{argsort}(\mathrm{argsort}(\mathrm{counts}))_i / \max\{N-1, 1\}$ with $q_0 \leftarrow \frac{1}{2}$ for padding: an ordinal ranking, not a log count, relative frequency, or tie-averaged quantile. Sorting is not explicitly stable, so tied counts can receive different ranks. BERT4Rec appends a zero-count mask-token slot before ranking, which changes N; the mask token receives its sorted rank. The learned scalars a and a_0 of Eq. 2 start at $a = \log(e-1)$ and $a_0 = 0$, so valid-token initial precision is $\min\{C, q_i + \log 2\}$ expanded over all features, and padding precision is zero.

Encoding. Every model starts with $h_t^0 = m_t D_p(\sqrt{d}\, E_{i_t} + P_t)$, where E and P are learned item and absolute-position embeddings and P_t is indexed by the position in the left-padded tensor. A valid-key mask permits $m_j = 1$; SASRec requires $j \le t$ in addition, BERT4Rec has no causal restriction and its mask token is a valid token. Hidden states are zeroed at padded queries after residual updates. Algorithm 1 gives the order, including the warm-up: for the first 50 optimizer steps the layer runs ordinary attention with standard dropout and no precision, gain, or distance bias, and at step 50 all three activate together, including the switch from ordinary attention dropout to survivor renormalization. Embedding precision and transport execute during warm-up but have no path to the loss. The statistics are evaluated on head outputs before padded-query masking; only e is masked at that point, and padding precision is reset at the end of the layer. Observation precision is capped before the gain and accumulated precision before transport. The warm-up counter is a Python attribute, not part of the saved model state; reconstructing a model from weights alone must explicitly restore its activation state.

Algorithm 1 Softmax PWA encoding (SASRec or terminal-masked BERT4Rec)

Require: IDs i, validity m, rank buffer q, completed optimizer steps s

Set $C = 100$, $\epsilon = 10^{-6}$, $\sigma_0^2 = 1/32$; choose causal or bidirectional mask M
$h_t \leftarrow m_t D_{0.5}(\sqrt{d}\, E_{i_t} + P_t)$; $\lambda_{tf} \leftarrow m_t \min\{C, \mathrm{softplus}(a) q_{i_t} + \mathrm{softplus}(a_0)\}$
for layers $\ell = 1, \dots, L$ **do**
  $x \leftarrow \mathrm{LN}_a(h)$; form headwise $q = W_Q x + b_Q$, $k = W_K h + b_K$, $v = W_V h + b_V$
  if $s < 50$ **then**
    $e \leftarrow m \odot \mathrm{ConcatHeads}(\mathrm{SDPA}(q, k, v, M, p = 0.5))$; $h^+ \leftarrow m \odot (h + e)$; $\lambda^+ \leftarrow \lambda$
  else
    $\beta_\ell \leftarrow -\log \max\{\mathrm{sigmoid}(r_\ell), 10^{-8}\}$; $s_{tj} \leftarrow q_t^\top k_j / \sqrt{d_k} - \beta_\ell |t - j|$; invalid logits $\leftarrow -\infty$
    $\alpha \leftarrow \mathrm{softmax}_j(s)$; replace NaNs by zero; $\alpha \leftarrow D_{0.5}(\alpha)$
    $\bar{\lambda}_j \leftarrow d_k^{-1} \sum_{f \in \text{head}} \lambda_{jf}$; $w_{tj} \leftarrow \alpha_{tj} \bar{\lambda}_j / (\sum_r \alpha_{tr} \bar{\lambda}_r + \epsilon)$
    $e_{tf} \leftarrow \sum_j w_{tj} v_{jf}$; $n_t \leftarrow 1/\max\{\sum_j w_{tj}^2, 10^{-8}\}$; $S_{tf} \leftarrow \max\{\sum_j w_{tj} v_{jf}^2 - e_{tf}^2, 0\}$
    $\lambda^{\mathrm{obs}} \leftarrow \mathrm{ConcatHeads}(\min\{C, n^2/(\sigma_0^2 + nS)\})$; $e \leftarrow m \odot \mathrm{ConcatHeads}(e)$
    $K \leftarrow \lambda^{\mathrm{obs}} / (\lambda + \lambda^{\mathrm{obs}} + \epsilon)$; $h^+ \leftarrow m \odot (h + K \odot e)$; $\lambda^+ \leftarrow \min\{C, \lambda + \lambda^{\mathrm{obs}}\}$
  end if
  $z \leftarrow \mathrm{LN}_f(h^+)$; $f \leftarrow D_{0.5}(W_2 D_{0.5}(\mathrm{ReLU}(W_1 z + b_1)) + b_2)$; $h \leftarrow m \odot (h^+ + f)$
  Compute j from Eq. 7 using the same z, W_1, W_2; compute λ by Appendix B.3
end for
return $\mathrm{LN}_{\mathrm{final}}(h) E^\top$

Non-unit and empty rows. With the additive guard, a non-empty row has weight mass $a_w = \sum_j w_j$ slightly below one, so the second-moment difference is not exactly a weighted residual sum: $\sum_j w_j (v_j - e)^2 = \sum_j w_j v_j^2 - e^2 + (a_w - 1) e^2$. An all-dropped or all-masked row has $w = 0$, $e = 0$, $S = 0$, $n = 10^8$, and observation precision C, not zero: a valid query with all entries dropped receives no hidden update but adds capped precision before transport. As an algebraic check, ideal weights $(\frac{1}{2}, \frac{1}{2})$, values $(2, 6)$, and $\sigma_0^2 = 1/32$ give $e = 4$, $S = 4$, $n = 2$, and uncapped observation precision $128/257$; a unit-weight single key gives 32 before capping.

C.2 HSTU: IDENTITY TRANSITION AND NATIVE-GATE MODULATION

HSTU has no separate feed-forward sublayer, so its prediction transition is the identity: $F_\ell(h^+) = h^+$, with Jacobian I, and $(h', \lambda') = (h^+, \lambda^+)$. In residual notation $F(h) = h + f(h)$ this means $f = 0$, not $f(h) = h$; the attention, native gate, and PWA update all remain nontrivial, and no process noise is added. HSTU uses a fused affine projection of $\mathrm{LN}_{\mathrm{in}}(h)$ into u_0, v, q, k and forms signed, unnormalized coefficients

$$A_{tj} = \mathbf{1}[M_{tj}] \, \frac{\mathrm{SiLU}(q_t^\top k_j / \sqrt{d_k})}{T} \, \exp(-\beta_\ell |t - j|), \tag{12}$$

where T includes padded positions, the distance factor is applied after the SiLU rather than as an additive score bias (an additive bias before SiLU would not implement the same factor, since

$\mathrm{SiLU}(s+b) \neq \mathrm{SiLU}(s)\exp(b)$), invalid scores receive the smallest finite value before SiLU and invalid coefficients are set to zero afterwards, and there is no attention-weight dropout. Precision enters the native gate through a single unconstrained scalar γ_ℓ per layer, initialized at zero:

$$u_{tf} = \mathrm{SiLU}(u_{0,tf})\ \exp\big(\mathrm{clip}(\gamma_\ell \log \max\{\lambda_{tf}, 10^{-6}\}, -5, 5)\big). \tag{13}$$

Aggregation uses $a = Av$. Because signed weights are not probabilities, the statistics use a separate normalized view, $\tilde{w}_{tj} = A_{tj}/(\sum_r A_{tr} + 10^{-6})$, $\tilde{e}_{tf} = \sum_j \tilde{w}_{tj} v_{jf}$, $S_{tf} = \max\{\sum_j \tilde{w}_{tj} v_{jf}^2 - \tilde{e}_{tf}^2, 0\}$, $n_t = 1/\max\{\sum_j \tilde{w}_{tj}^2, 10^{-8}\}$, and the regularized proxy $\lambda_{tf}^{\mathrm{obs}} = m_t \min\{100, (n_t + 1)/(S_{tf} + 1/32)\}$, which is not Eq. 5. Normalizing by a signed sum does not produce a probability distribution, so n need not be at least one and S need not be non-negative before clipping; for instance normalized weights $(2, -1)$ and values $(0, 1)$ give $n = 1/5$, $\tilde{e} = -1$, and a second-moment difference of -2, clipped to zero. The ratio is therefore an empirical reliability proxy without established calibration. The residual gate scales the entire projected output $o = m \odot (D_{0.2}([u, h, u \odot \mathrm{LN}_{\mathrm{attn}}(a)])W_{\mathrm{out}} + b_{\mathrm{out}})$ through K as in Eq. 6, but accumulated HSTU precision is not capped and can exceed C. Observation precision is measured in pre-projection value coordinates and is not transported through the unnormalized aggregation scale, the attention LayerNorm, the gate, the concatenation, dropout, or the output projection. Initialization uses the frozen count-based rule $\lambda_{tf}^0 = m_t \max\{4.5 q_{i_t}, \epsilon\} + (1 - m_t)$ with no learned range or offset, and HSTU has no warm-up.

C.3 What the baseline comparison identifies

The local softmax baselines form attention and feed-forward residuals with normalized carriers, $r_{\mathrm{base}} = \mathrm{LN}_a(h) + e$ and $h'_{\mathrm{base}} = \mathrm{LN}_f(r_{\mathrm{base}}) + f(\mathrm{LN}_f(r_{\mathrm{base}}))$, whereas PWA carries $r_{PWA} = h + K \odot e$ and $h'_{PWA} = r_{PWA} + f(\mathrm{LN}_f(r_{PWA}))$. Setting $K = 1$ does not remove this difference, and precision reweighting, the distance bias, and training-time renormalization can remain active. The base-versus-PWA comparison therefore evaluates a composite construction, which is why the residual-, dropout-, and parameter-matched carrier control of Section 5.5 is needed for component-level attribution. No precision likelihood, calibration objective, or terminal precision term appears in the loss; the last layer's post-transition precision cannot affect the logits, and returned precision should not be read as calibrated covariance of the final representation.

D Data, effective configuration, and provenance

Table 5: Processed dataset statistics from the recovered leave-one-out splits, after iterative five-core filtering. Interaction counts include training, validation, and test events; mean length is the mean number of interactions per user; density is interactions divided by users times items, computed from the columns shown. $T_{\max}$ is the configured total context size, fixed per dataset at one position above the 99th percentile of evaluation-input history length; BERT4Rec reserves one slot for its mask token.

Dataset	Users	Items	Interactions	Mean length	Density	$T_{\max}$
Instruments	1,429	900	10,261	7.2	0.80%	23
Sports	35,598	18,357	296,337	8.3	0.05%	32
Toys	19,412	11,924	167,597	8.6	0.07%	40
Beauty	22,363	12,101	198,502	8.9	0.07%	43
Games	24,303	10,672	231,780	9.5	0.09%	51
ML-1M	6,040	3,416	999,611	165.5	4.85%	906

Context limits. We fix $T_{\max}$ per dataset rather than at one shared constant because a constant generous for ML-1M would be wasted padding on Instruments, and one sized for the Amazon categories would truncate a large fraction of ML-1M histories and confound the history-length reading of Section 5.3. Under this rule truncation touches at most the longest 1% of histories on every dataset; a script checks the configured values against the processed splits and fails on any departure. Because $T_{\max}$ is part of the arm definition, runs at a different limit are not pooled with these.

Information boundary. The count source consists of complete, untruncated training histories plus each validation-target identity; test targets are not included. Validation identities can therefore influence representation initialization and attention before checkpoint selection. A strictly train-only protocol would be a different comparison, not a relabeling of this one.

Objectives and evaluation. SASRec and HSTU use categorical next-item cross-entropy at all valid target positions. BERT4Rec replaces only the last retained training item by [MASK] and supervises that position, in both arms; there is no random-Cloze masking or multi-truncation augmentation. At evaluation, appending [MASK] reserves one context slot, so BERT4Rec retains at most $T_{\max} - 1$ real items while the causal encoders retain $T_{\max}$. Training scores the complete tied embedding vocabulary, ignoring padding as a target but retaining it as a negative class; evaluation masks padding and seen input IDs. All models score $\mathrm{LN}_{\mathrm{final}}(h^L)E^\top$. The backbone grid selects the checkpoint with the highest validation NDCG@10, evaluated every 5 epochs; patience 20 counts evaluations without improvement, and training allows up to 200 epochs. The cohort analysis independently recomputes metrics from per-user rank records and fails if a rank record and the primary result disagree beyond floating-point aggregation error.

Optimization. Adam with base learning rate 10^{-3}, batch size 256, global gradient-norm clipping at 5, no scheduler. The item-embedding matrix receives weight decay 0.01; other parameters receive none. For the softmax encoders the global range logit a and the distance logits r_ℓ use learning rate 10^{-1}, and the offset logit a_0 and the backbone weights 10^{-3}; HSTU routes all parameters, including γ_ℓ and r_ℓ, through 10^{-3}. The requested HSTU warm-up and learning-rate multiplier are not effective in the audited runner, and its effective dropout is 0.2; these are implementation facts, not sensitivity results. Table 6 lists the effective recipes.

Table 6: Effective PWA recipes. Requested values come from recovered checkpoints and the recorded launcher; effective HSTU overrides and optimizer dispatch are from the audited runner.

Setting	SASRec+PWA	BERT4Rec+PWA	HSTU+PWA
Hidden width / layers / heads	64 / 2 / 2	64 / 2 / 2	64 / 2 / 2
FFN width	256	256	none
Effective dropout	0.5	0.5	0.2
Precision cap C	100	100	100 (observations only)
Input prior	learned range/offset	learned range/offset	fixed count rank
Observation estimator	posterior form (Eq. 5)	posterior form (Eq. 5)	regularized proxy
Transport	diagonal Jacobian	diagonal Jacobian	identity
Position bias	additive logit, ρ_ℓ init 0.99	additive logit, ρ_ℓ init 0.99	post-SiLU multiplier
Effective warm-up	50 steps	50 steps	none
Base learning rate / batch	10^{-3} / 256	10^{-3} / 256	10^{-3} / 256
Distance-logit learning rate	0.1	0.1	0.001
Range / offset logit learning rate	0.1 / 0.001	0.1 / 0.001	fixed / none
Max epochs / patience (evaluations)	200 / 20	200 / 20	200 / 20

Seeds. Every cell uses the same 20 training seeds, the consecutive integers 42 to 61, for the baseline arm, the PWA arm, and the STOSA reference; the attribution and sensitivity arms use the first five of them.

Evidence and its limits. The artifact inventory separates the complete backbone grid, the official-code STOSA reference, and exploratory or obsolete arms. Numerical aggregation requires the exact seed set, rejects missing or duplicated runs and non-finite metrics, and preserves the identity of each input file. Representative checkpoints recover model configurations and precision buffers; they do not recover every optimizer setting or an immutable historical source tree.

E Parameter and wall-clock cost

The counts and ratios in Table 7 are generated from the run registry, and the parameter deltas are measured by constructing each backbone with and without PWA at the evaluated configuration. The headline parameter delta overstates what the evaluated configuration trains: the recipe sets the

process-noise ceiling to zero, and gradient introspection finds only 4 of the 132 SASRec additions receive gradient, the remaining 128 process-noise logits being inert. The wall-clock overhead falls on the sequence encoder, so it shrinks as the output softmax grows with catalogue size.

Table 7: Cost of PWA. Added parameters are independent of catalogue size, so the relative increase shrinks as the catalogue grows. Wall-clock ratios are the median over datasets of the per-dataset median PWA:baseline training time, matched on 20 seeds; the range across datasets is in brackets.

Backbone	Added params	Added %	Train time	Range
SASRec	132	0.0031–0.0875%	2.12×	1.38–2.39×
BERT4Rec	132	0.0031–0.0874%	1.83×	1.26–2.03×
HSTU	4	0.0001–0.0034%	1.14×	1.06–1.80×

F COMPLETE BACKBONE COMPARISONS

Table 8: All 54 base-vs-PWA paired tests (18 cells × 3 metrics): means, standard deviations, absolute and relative differences, pointwise 95% paired-difference intervals, seeds in favor, raw and Holm-adjusted two-sided paired-t p-values. Holm correction spans the entire family.

Dataset	Backbone	Metric	Base μ	Base σ	PWA μ	PWA σ	Δ	$\%\Delta$	CI low	CI high	wins/n	raw p	Holm p
Instruments	SASRec	HR@10	0.1169	0.0036	0.1220	0.0054	0.0051	4.34	0.0028	0.0073	18/20	1.7e-04	3.2e-03
Instruments	SASRec	NDCG@10	0.0649	0.0016	0.0680	0.0025	0.0031	4.74	0.0020	0.0042	18/20	9.5e-06	2.7e-04
Instruments	SASRec	MRR	0.0590	0.0016	0.0615	0.0020	0.0025	4.24	0.0013	0.0037	18/20	3.7e-04	5.9e-03
Instruments	BERT4Rec	HR@10	0.0783	0.0041	0.0801	0.0040	0.0018	2.32	−0.0008	0.0044	15/20	1.6e-01	1.0e+00
Instruments	BERT4Rec	NDCG@10	0.0427	0.0019	0.0435	0.0024	0.0007	1.74	−0.0007	0.0022	12/20	2.9e-01	1.0e+00
Instruments	BERT4Rec	MRR	0.0406	0.0015	0.0409	0.0022	0.0003	0.78	−0.0010	0.0017	10/20	6.3e-01	1.0e+00
Instruments	HSTU	HR@10	0.0946	0.0049	0.0985	0.0054	0.0040	4.18	0.0012	0.0067	16/20	7.4e-03	7.4e-02
Instruments	HSTU	NDCG@10	0.0546	0.0029	0.0580	0.0027	0.0034	6.23	0.0020	0.0048	18/20	5.6e-05	1.4e-03
Instruments	HSTU	MRR	0.0505	0.0028	0.0538	0.0024	0.0033	6.46	0.0021	0.0044	18/20	1.3e-05	3.5e-04
Sports	SASRec	HR@10	0.0491	0.0009	0.0529	0.0008	0.0038	7.68	0.0032	0.0043	20/20	9.5e-12	4.2e-10
Sports	SASRec	NDCG@10	0.0268	0.0007	0.0292	0.0006	0.0024	8.80	0.0020	0.0027	20/20	1.1e-11	4.7e-10
Sports	SASRec	MRR	0.0251	0.0007	0.0271	0.0005	0.0020	7.84	0.0017	0.0023	20/20	4.1e-11	1.7e-09
Sports	BERT4Rec	HR@10	0.0260	0.0008	0.0280	0.0008	0.0020	7.62	0.0015	0.0025	19/20	1.3e-07	4.3e-06
Sports	BERT4Rec	NDCG@10	0.0146	0.0007	0.0155	0.0005	0.0009	5.87	0.0005	0.0012	16/20	8.7e-05	1.7e-03
Sports	BERT4Rec	MRR	0.0143	0.0007	0.0149	0.0005	0.0006	4.52	0.0003	0.0010	16/20	1.4e-03	1.7e-02
Sports	HSTU	HR@10	0.0437	0.0010	0.0433	0.0015	−0.0004	−1.00	−0.0011	0.0003	7/20	2.1e-01	1.0e+00
Sports	HSTU	NDCG@10	0.0244	0.0005	0.0243	0.0007	−0.0002	−0.65	−0.0005	0.0002	9/20	3.5e-01	1.0e+00
Sports	HSTU	MRR	0.0229	0.0005	0.0228	0.0006	−0.0001	−0.61	−0.0005	0.0002	8/20	3.8e-01	1.0e+00
Toys	SASRec	HR@10	0.0869	0.0015	0.0919	0.0013	0.0050	5.73	0.0042	0.0058	20/20	4.8e-11	1.9e-09
Toys	SASRec	NDCG@10	0.0508	0.0011	0.0542	0.0007	0.0034	6.71	0.0028	0.0040	20/20	1.1e-10	4.3e-09
Toys	SASRec	MRR	0.0460	0.0010	0.0491	0.0007	0.0030	6.62	0.0025	0.0036	20/20	2.3e-10	8.3e-09
Toys	BERT4Rec	HR@10	0.0332	0.0013	0.0364	0.0013	0.0032	9.64	0.0022	0.0042	20/20	1.5e-06	4.5e-05
Toys	BERT4Rec	NDCG@10	0.0197	0.0010	0.0214	0.0008	0.0017	8.57	0.0010	0.0024	17/20	7.2e-05	1.6e-03
Toys	BERT4Rec	MRR	0.0185	0.0010	0.0200	0.0008	0.0016	8.45	0.0009	0.0022	16/20	8.2e-05	1.7e-03
Toys	HSTU	HR@10	0.0677	0.0016	0.0702	0.0016	0.0026	3.79	0.0014	0.0037	16/20	2.1e-04	3.6e-03
Toys	HSTU	NDCG@10	0.0421	0.0013	0.0436	0.0014	0.0016	3.75	0.0007	0.0025	16/20	1.3e-03	1.7e-02
Toys	HSTU	MRR	0.0388	0.0012	0.0402	0.0014	0.0014	3.49	0.0005	0.0022	15/20	2.6e-03	2.9e-02
Beauty	SASRec	HR@10	0.0881	0.0013	0.0901	0.0015	0.0020	2.31	0.0012	0.0029	19/20	5.6e-05	1.4e-03
Beauty	SASRec	NDCG@10	0.0501	0.0008	0.0513	0.0009	0.0012	2.42	0.0007	0.0017	18/20	6.4e-05	1.5e-03
Beauty	SASRec	MRR	0.0450	0.0007	0.0462	0.0008	0.0012	2.60	0.0007	0.0016	18/20	1.8e-05	4.7e-04
Beauty	BERT4Rec	HR@10	0.0428	0.0009	0.0470	0.0009	0.0043	9.95	0.0036	0.0049	20/20	8.3e-12	3.7e-10
Beauty	BERT4Rec	NDCG@10	0.0240	0.0006	0.0261	0.0006	0.0022	9.10	0.0018	0.0026	20/20	1.6e-10	6.1e-09
Beauty	BERT4Rec	MRR	0.0219	0.0006	0.0238	0.0005	0.0019	8.70	0.0015	0.0023	20/20	4.7e-09	1.6e-07
Beauty	HSTU	HR@10	0.0713	0.0018	0.0727	0.0015	0.0014	1.98	0.0001	0.0027	16/20	3.8e-02	3.4e-01
Beauty	HSTU	NDCG@10	0.0429	0.0009	0.0433	0.0011	0.0004	0.93	−0.0004	0.0012	13/20	3.1e-01	1.0e+00
Beauty	HSTU	MRR	0.0393	0.0007	0.0396	0.0010	0.0003	0.74	−0.0004	0.0010	12/20	3.6e-01	1.0e+00
Games	SASRec	HR@10	0.1346	0.0021	0.1382	0.0019	0.0036	2.67	0.0024	0.0048	18/20	5.3e-06	1.5e-04
Games	SASRec	NDCG@10	0.0722	0.0011	0.0749	0.0012	0.0027	3.68	0.0020	0.0034	20/20	2.1e-07	6.7e-06
Games	SASRec	MRR	0.0641	0.0010	0.0665	0.0011	0.0025	3.85	0.0018	0.0031	19/20	3.0e-07	9.3e-06
Games	BERT4Rec	HR@10	0.0729	0.0020	0.0794	0.0011	0.0066	9.02	0.0056	0.0076	20/20	3.1e-11	1.3e-09
Games	BERT4Rec	NDCG@10	0.0384	0.0011	0.0417	0.0006	0.0033	8.53	0.0027	0.0038	20/20	1.6e-10	6.1e-09
Games	BERT4Rec	MRR	0.0351	0.0011	0.0377	0.0005	0.0026	7.41	0.0021	0.0031	20/20	1.4e-09	4.9e-08
Games	HSTU	HR@10	0.1221	0.0020	0.1247	0.0016	0.0027	2.20	0.0015	0.0039	18/20	1.7e-04	3.2e-03
Games	HSTU	NDCG@10	0.0670	0.0011	0.0683	0.0008	0.0013	1.93	0.0006	0.0020	17/20	7.7e-04	1.2e-02
Games	HSTU	MRR	0.0601	0.0009	0.0611	0.0006	0.0011	1.78	0.0005	0.0016	15/20	8.4e-04	1.2e-02
ML-1M	SASRec	HR@10	0.2341	0.0036	0.2524	0.0027	0.0183	7.81	0.0164	0.0202	20/20	3.9e-14	1.9e-12
ML-1M	SASRec	NDCG@10	0.1302	0.0022	0.1442	0.0014	0.0140	10.72	0.0126	0.0153	20/20	7.0e-15	3.8e-13
ML-1M	SASRec	MRR	0.1133	0.0019	0.1258	0.0012	0.0126	11.09	0.0113	0.0138	20/20	7.0e-15	3.8e-13
ML-1M	BERT4Rec	HR@10	0.0665	0.0037	0.0943	0.0041	0.0278	41.72	0.0251	0.0304	20/20	7.2e-15	3.8e-13
ML-1M	BERT4Rec	NDCG@10	0.0365	0.0025	0.0513	0.0022	0.0148	40.65	0.0131	0.0165	20/20	2.1e-13	1.0e-11
ML-1M	BERT4Rec	MRR	0.0343	0.0024	0.0472	0.0018	0.0129	37.73	0.0114	0.0145	20/20	4.7e-13	2.2e-11
ML-1M	HSTU	HR@10	0.2712	0.0041	0.2909	0.0035	0.0197	7.28	0.0174	0.0221	20/20	3.1e-13	1.5e-11
ML-1M	HSTU	NDCG@10	0.1566	0.0023	0.1703	0.0026	0.0137	8.76	0.0123	0.0151	20/20	2.6e-14	1.3e-12
ML-1M	HSTU	MRR	0.1358	0.0019	0.1478	0.0027	0.0120	8.81	0.0106	0.0133	20/20	1.4e-13	6.9e-12

G STOSA COMPARISON: PROTOCOL AND FULL STATISTICS

The STOSA reference preserves the released model computation at revision `1f00ecb193a7ea7fe1bc4d14e999ce4ebec845c2`: separate mean and covariance embeddings, Wasserstein-based attention and ranking, its native pairwise objective with the auxiliary PVN term, and a separately selected configuration, so the contrast is not capacity- or objective-matched with the backbone grid. STOSA selects its checkpoint by validation MRR@40 at every epoch (patience 20 epochs); the backbone grid validates NDCG@10 every five epochs and counts patience in evaluation events, so equal epoch limits and patience arguments do not imply equal stopping budgets. STOSA masks previously seen items using its full-history rating matrix while the backbone grid masks items in the retained input context; these exclusions can differ for histories longer than the context limit. The native STOSA candidate list retains its reserved embedding slots, whereas our grid excludes padding and the appended mask token. The native log field named MRR is truncated: the released trainer passes only its forty best candidates to the reciprocal-rank routine and assigns zero when the target is absent, so we compute the same truncated metric for PWA from per-user ranks and label the comparison MRR@40; no full-MRR value is inferred for STOSA. The released encoder discards the transformed mean tensor returned by its attention block while propagating the covariance branch; the input mean embeddings still learn. We preserve this released behavior rather than substituting a modified implementation and do not attribute any gap to it. Table 9 gives the full statistics behind Table 3.

Table 9: STOSA vs. SASRec+PWA, 18 paired tests on 20 matched seeds. Statistics use recorded values before rounding; all 18 raw p-values are below 10^{-14}, so every comparison survives Holm correction over this family at $p < 0.0001$.

Dataset	Metric	STOSA μ	STOSA σ	PWA μ	PWA σ	Δ	%Δ	CI low	CI high	wins/n	raw p
Instruments	HR@10	0.0795	0.0054	0.1220	0.0054	0.0425	53.50	0.0387	0.0463	20/20	1.9e-15
Instruments	NDCG@10	0.0410	0.0036	0.0680	0.0025	0.0270	65.71	0.0249	0.0291	20/20	1.2e-16
Instruments	MRR@40	0.0343	0.0036	0.0574	0.0021	0.0232	67.51	0.0211	0.0252	20/20	1.3e-15
Sports	HR@10	0.0340	0.0017	0.0529	0.0008	0.0189	55.69	0.0181	0.0197	20/20	1.1e-21
Sports	NDCG@10	0.0186	0.0011	0.0292	0.0006	0.0106	57.18	0.0101	0.0111	20/20	1.9e-20
Sports	MRR@40	0.0160	0.0010	0.0250	0.0005	0.0090	56.02	0.0085	0.0094	20/20	1.0e-19
Toys	HR@10	0.0705	0.0024	0.0919	0.0013	0.0214	30.40	0.0202	0.0226	20/20	2.8e-19
Toys	NDCG@10	0.0417	0.0016	0.0542	0.0007	0.0126	30.14	0.0117	0.0134	20/20	6.6e-18
Toys	MRR@40	0.0358	0.0015	0.0468	0.0007	0.0109	30.51	0.0101	0.0117	20/20	3.3e-17
Beauty	HR@10	0.0631	0.0026	0.0901	0.0015	0.0270	42.74	0.0256	0.0284	20/20	7.0e-20
Beauty	NDCG@10	0.0355	0.0017	0.0513	0.0009	0.0158	44.61	0.0150	0.0167	20/20	1.4e-19
Beauty	MRR@40	0.0304	0.0015	0.0438	0.0008	0.0134	44.11	0.0127	0.0141	20/20	1.2e-19
Games	HR@10	0.0894	0.0025	0.1382	0.0019	0.0488	54.54	0.0472	0.0503	20/20	6.4e-24
Games	NDCG@10	0.0461	0.0013	0.0749	0.0012	0.0287	62.27	0.0278	0.0296	20/20	5.9e-24
Games	MRR@40	0.0387	0.0010	0.0630	0.0011	0.0243	62.93	0.0235	0.0251	20/20	9.8e-24
ML-1M	HR@10	0.1588	0.0054	0.2524	0.0027	0.0936	58.97	0.0912	0.0961	20/20	1.9e-25
ML-1M	NDCG@10	0.0857	0.0029	0.1442	0.0014	0.0584	68.18	0.0570	0.0599	20/20	3.8e-26
ML-1M	MRR@40	0.0724	0.0026	0.1220	0.0012	0.0496	68.54	0.0484	0.0509	20/20	7.8e-26

H COHORT PROTOCOL

Item-exposure cohorts group test targets by their pre-test catalogue frequency; history cohorts group users by the full history available before the test target. The same cohort assignment is used for both arms. Each axis is split at its median into two cohorts; nominal thresholds keep ties together, targets with exactly zero pre-test exposure form a separate cohort that is not part of either half (it exists on the six Amazon datasets, where both arms score exactly zero on HR@10 and NDCG@10 in that cohort on every seed, and PWA is slightly worse there on MRR on all six, which we report rather than omit: the method needs some observed exposure before its precision estimate can help), and empty cohorts are not silently treated as zero-valued observations. For each cohort, user-averaged metrics are computed separately in every seed; the aggregate relative gain is the ratio of unweighted dataset-macro means, not a user-pooled metric or an average of unstable per-user ratios. Absolute gains are retained in the analysis artifacts, since a large relative change can result from a small baseline denominator. Paired seed resampling supplies the uncertainty of the cohort contrasts, and stars in Table 2 are Holm-adjusted within each block's family; raw p-values are in `metadata/stratification_cells.csv`. The % pop. column of Table 2 comes from labelling every scored test interaction on both axes at once: for each user cohort it is the share of that

cohort's held-out items that fall in the popular half of the catalogue, computed from the joint cohort counts. Its values (short-history half first, long-history half second) are Instruments 56.3 and 51.5, Sports 66.5 and 65.6, Toys 66.2 and 65.6, Beauty 65.7 and 67.3, Games 77.5 and 76.2, ML-1M 90.5 and 82.5. The two axes are therefore confounded in the same direction on five of six datasets, most strongly on ML-1M, which is why Section 5.3 reads the history-length gradient as a contrast within the popular-item regime rather than as an independent cold-start effect. These analyses are observational and conditional on the benchmark preprocessing; cohort membership is not randomly assigned.

I SENSITIVITY VALUES

Table 10: Values behind Figure 2: NDCG@10 on SASRec when one setting is swept and the rest held at the headline recipe, over 5 matched seeds. The headline setting (–) is the published run rather than a re-run, so each curve passes through the value of Table 1; Δ is relative to it on the same seeds. No Holm-adjusted paired-t test within this table's family reaches $p < 0.05$.

		Beauty		ML-1M	
Parameter	Value	NDCG@10	Δ	NDCG@10	Δ
Precision cap C	10	0.0508	-1.93%	0.1432	-0.32%
	30	0.0504	-2.73%	0.1427	-0.67%
	100	0.0518	–	0.1436	–
	300	0.0518	-0.04%	0.1433	-0.21%
	1000	0.0522	$+0.71\%$	0.1439	$+0.20\%$
Precision warm-up (steps)	0	0.0513	-0.98%	0.1447	$+0.71\%$
	25	0.0517	-0.29%	0.1445	$+0.62\%$
	50	0.0518	–	0.1436	–
	100	0.0521	$+0.64\%$	0.1445	$+0.60\%$
	200	0.0522	$+0.69\%$	0.1451	$+1.00\%$
Position-bias init ρ_ℓ	0.9	0.0517	-0.16%	0.1437	$+0.07\%$
	0.95	0.0510	-1.66%	0.1428	-0.58%
	0.99	0.0518	–	0.1436	–
	0.999	0.0520	$+0.32\%$	0.1456	$+1.35\%$
	0.9999	0.0523	$+0.85\%$	0.1464	$+1.95\%$