EXPLAINABILITY-GUIDED HISTORY DISTILLATION FOR EFFICIENT SEQUENTIAL RECOMMENDATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Sequential recommendation models, spanning self-attention based Transformer architectures to state-space models, rely on processing complete interaction histories, incurring inference costs that scale with sequence length. While efficient attention variants and heuristic truncation mitigate this computational footprint, they overlook internal model attribution signals and neglect the efficiency gains achievable by distilling the raw interaction sequence itself. To bridge this gap, we propose explainability-driven sequence distillation, a retraining-free framework that leverages post-hoc attribution methods to score each interaction's predictive contribution. By retaining only the highest-scoring positions, our framework compresses user histories offline, enabling shorter sequences to accelerate online inference without altering underlying model parameters. Across four dataset configurations and four architectural paradigms, including autoregressive, bidirectional, state-space, and long-sequence backbones, Integrated Gradient-based sequence distillation preserves or improves ranking quality while substantially reducing inference GFLOPs, outperforming recency-based truncation by an average of 11.5% across all evaluated settings. Crucially, even on benchmarks with extreme temporal recency bias, our approach matches or exceeds recency truncation in the performance-efficiency trade-off, an advantage that widens under noisy input sequences. These findings demonstrate that post-hoc explainability, traditionally restricted to model interpretation, serves as a general, retraining-free principle for distilling raw input contexts and reducing online serving costs in sequential recommendation.

1 INTRODUCTION

Sequential recommendation models predict which item a user will interact with next, given the chronological sequence of their past interactions. This formulation is central to modern recommender systems: e-commerce platforms, streaming services, and short-video applications all rely on a user's recent behavior to rank candidate items for their next session. The input to these models is a sequence of item identifiers, ordered by time, where each identifier corresponds to a past user interaction, such as a purchase, a view, or a click, truncated or padded to a fixed context window of length L. Each item is mapped to a learned embedding vector, and the model processes the resulting sequence of embeddings to produce a score over the item catalog. Modern architectures such as SASRec (Kang & McAuley, 2018) and BERT4Rec (Sun et al., 2019) implement this mapping using self-attention, allowing every position in the sequence to directly attend to every other position, which has proven highly effective at capturing long-range dependencies in user behavior.

This effectiveness comes at a cost. Self-attention scales quadratically with sequence length L: both memory and inference compute grow as $\mathcal{O}(L^2)$. As platforms retain longer interaction histories, whether because users have been active for years or because a domain naturally produces high-frequency interactions Katariya et al. (2025), this quadratic scaling becomes a deployment bottleneck. Serving predictions at low latency and low cost is a practical requirement, and naively feeding an ever-growing history into a self-attention model is increasingly incompatible with that requirement.

A natural way to reduce this cost is to shorten the sequence before inference (Katariya et al., 2025). Efficient attention variants, such as linear (Katharopoulos et al., 2020) and sparse (Child et al., 2019) attention, reduce the cost of processing a fixed-length sequence, but they do not decide which interactions to keep when the sequence itself must be truncated. Simple heuristics fill this gap.

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.

Recency-based truncation keeps only the most recent K interactions. Random truncation keeps K interactions at random. Neither heuristic uses any signal from the trained model, and both implicitly assume that recency, or arbitrary coverage, is a good proxy for relevance. This assumption does not hold universally: prior work shows that sequential recommendation models concentrate their predictive signal on a small, dataset-dependent subset of interactions (Rao et al., 2021; He et al., 2021), and, as we show later, this subset is not always the most recent one.

Storing and caching interaction data for online inference is computationally expensive and memory-intensive, particularly as sequence lengths scale. This raises a natural question: can we identify the informative subset of interactions using signals from the model itself, without retraining it? Post-hoc explainability methods are designed exactly for this purpose. Given a trained model and an input, they assign a score to each input position reflecting its contribution to the prediction. By computing these scores offline to prune interaction histories before deployment, we pay the explanation cost once in storage. Online inference then operates on significantly shorter sequences, simultaneously drastically reducing inference GFLOPs.

Our Contributions. We introduce explainability-guided sequence distillation, a **model-agnostic**, **retraining-free framework** that integrates post-hoc attribution into the inference pipeline of sequential recommendation models. Rather than serving full interaction histories, our framework retains only the highest-scoring positions identified by an explainability method, **compressing sequences offline** to drastically **accelerate online inference** without altering underlying model parameters. Our framework seamlessly supports any post-hoc attribution method that yields position-wise feature importance and generalizes across diverse sequential recommendation architectures. Our key contributions are:

- We formalize sequence distillation as an architecture-agnostic position selection problem guided by per-step attribution scores, introducing a normalized AUC metric to quantify performance across the compression spectrum alongside Pareto trade-off analyses at optimal operating points.
- We perform an extensive empirical evaluation across four backbone architectures and four dataset configurations, demonstrating that attribution-guided distillation consistently outperforms standard recency truncation and random pruning, achieving superior performance-efficiency trade-offs.
- Through comprehensive ablations, we demonstrate the robustness of explainability-guided distillation under sequence corruption, while characterizing its sensitivity to model capacity and distinct attention mechanisms (standard, linear, and sparse).

2 Related Works

Architecture-Based Efficient Models. Mitigating the quadratic complexity $\mathcal{O}(L^2)$ of self-attention in sequential recommendation has driven extensive architectural research. Approaches include linearizing attention via kernel operations to achieve $\mathcal{O}(L)$ complexity (Katharopoulos et al., 2020), restricting attention to localized or strided windows via sparse patterns (Child et al., 2019), and projecting context dimensions via low-rank approximations (Wang et al., 2020). More recently, selective state space models (SSMs) like Mamba have emerged as linear-time alternatives to Transformers for long user sequences (Gu & Dao, 2023). However, while these structural modifications reduce the compute footprint per step, they process fixed context windows uniformly and cannot identify or prune uninformative interactions from the input history itself.

Patching in Time Series and Sequential Modeling. Patching compresses sequences by grouping contiguous time steps into local sub-sequences. Originally popularized in time series forecasting to preserve temporal structure (e.g., PatchTST (Nie et al., 2022)), patching has recently been adapted to sequential recommendation, particularly with Large Language Model (LLM) backbones such as PatchRec, which aggregates textual item tokens and sessions into multi-grained patches directly at the raw input prompt level during online inference (Liao et al., 2025). However, for complex temporal user interactions comprising a mixture of categorical and numerical features, such patching must typically be performed online in the latent feature space, leaving the domain of raw input-level sequence distillation unexplored.

Crucially, raw sequence-level distillation is orthogonal to architectural efficiency and can be seamlessly combined with the models discussed above; we empirically demonstrate this synergy in our results by evaluating our proposed framework across standard, linear, and sparse attention mechanisms.

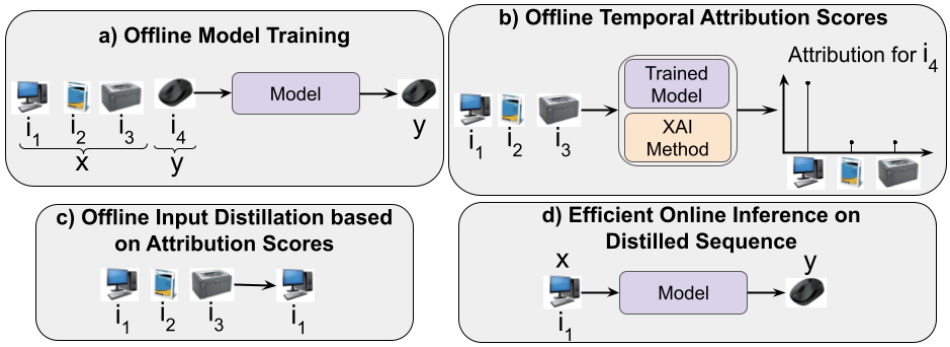


Figure 1: Framework pipeline. (a) Offline training on full history. (b) Post-hoc attribution scoring using a trained model and XAI method. (c) Offline sequence distillation retaining top-attributed items (i_1). (d) Efficient online inference on distilled history with the frozen model.

3 METHODOLOGY

3.1 PROBLEM FORMULATION

Let $\mathcal{U}$ and $\mathcal{I}$ denote the user and item sets, with $|\mathcal{U}|$ users and $|\mathcal{I}|$ items. For a user $u \in \mathcal{U}$, we observe a chronologically ordered interaction sequence $\mathcal{S}_u = (i_1, \ldots, i_L)$ truncated to a fixed context window of length L. An embedding matrix $\mathbf{E} \in \mathbb{R}^{|\mathcal{I}| \times d}$ maps each item to a d-dimensional vector $\mathbf{e}_t \in \mathbb{R}^d$, yielding the input representation

$$\mathbf{X} = \left[\mathbf{e}_1; , \mathbf{e}_2; , \ldots ; , \mathbf{e}_L\right] \in \mathbb{R}^{L \times d}. \tag{1}$$

Depending on the dataset, $\mathbf{E}$ may encode raw item features (e.g., genre tags, interaction counts) or serve as a learnable lookup table trained end-to-end with the model. A sequential recommendation model $f_\theta : \mathbb{R}^{L \times d} \to \mathbb{R}^\gamma$ maps the input sequence to a score vector $\hat{\mathbf{y}} \in \mathbb{R}^\gamma$. For binary or multi-class prediction tasks (e.g., whether a user will engage with a given video), $\gamma = C$ where C is the number of classes; for next-item prediction, $\gamma = |\mathcal{I}|$.

Not all positions in $\mathcal{S}_u$ contribute equally to the model's decision. Prior work has shown that sequential recommendation models concentrate their predictive signal on a small subset of interactions (Rao et al., 2021; He et al., 2021). In practice, interaction histories are stored offline and served at inference time; this raises a natural question: *can we distill the stored interaction sequence offline, prior to deployment, using post-hoc explainability methods, such that only the most informative time steps are retained for online inference, reducing computational cost without retraining the model?*

To this end, we employ a post-hoc explanation method $g(f_\theta, \mathbf{X}, c) \to \boldsymbol{\phi} \in \mathbb{R}^L$ that assigns a scalar attribution score ϕ_j to each position $j \in [1, L]$ with respect to a target class or item c. A positive ϕ_j indicates that the interaction at time step j increases the predicted probability $\hat{y}_c$, while a negative ϕ_j suppresses it. Crucially, this attribution-guided distillation is performed strictly offline; during online serving, newly arriving interaction data is appended directly to the sequence and subsequently distilled in the next offline cycle. We empirically demonstrate that retaining only the high-attribution time steps either preserves predictive performance or, in many cases, improves it by removing noisy and irrelevant interactions, while simultaneously reducing GFLOPs. The complete end-to-end framework, from offline attribution scoring to efficient online inference, is illustrated in Figure 1.

3.2 XAI-DRIVEN DISTILLATION OF SEQUENCE

Step 1: Training on the full sequence. We train a transformer-based sequential recommendation model f_θ on the full interaction sequence $\mathbf{X} \in \mathbb{R}^{L \times d}$ with context window length L kept intact. Concretely, we consider SASRec (Kang & McAuley, 2018) and BERT4Rec (Sun et al., 2019) as representative architectures, both of which employ multi-head self-attention over the full sequence.

The computational cost of a single forward pass scales as $\mathcal{O}(L^2 d)$ in the attention layers, making inference latency sensitive to the sequence length L.

Step 2: Computing temporal attribution scores. Once f_θ is trained and frozen, we apply a post-hoc explanation method $g(f_\theta, \mathbf{X}, c) \to \boldsymbol{\phi} \in \mathbb{R}^L$, where ϕ_j quantifies the contribution of the interaction at position $j \in [1, L]$ to the predicted score $\hat{y}_c$, with $c = \arg\max_c \hat{\mathbf{y}}$ being the top predicted item and $\hat{\mathbf{y}} \in \mathbb{R}^{|\mathcal{I}|}$ the predicted score vector over all items.

Step 3: Distilling the sequence. Given attribution scores $\boldsymbol{\phi} \in \mathbb{R}^L$, we select the indices of the top-K highest attribution scores and sort them chronologically to preserve causal ordering:

$$\mathcal{S}_K = \text{sort}\left(\text{ArgTopK}(\boldsymbol{\phi}, K)\right), \qquad K \in \{1, \ldots, L\}. \tag{2}$$

The distilled input sequence $\mathbf{X}_{\mathcal{S}_K} \in \mathbb{R}^{K \times d}$ is then constructed by selecting the corresponding vectors:

$$\mathbf{X}_{\mathcal{S}_K} = [\mathbf{x}_j]_{j \in \mathcal{S}_K}, \tag{3}$$

which retains only the K most informative interactions. Note that Steps 2 and 3 are executed offline during user history storage, enabling efficient online inference.

Step 4: Inference on the distilled sequence. During the online inference stage, the frozen model f_θ is evaluated directly on $\mathbf{X}_{\mathcal{S}_K}$:

$$\hat{\mathbf{y}}_{\mathcal{S}_K} = f_\theta\left(\mathbf{X}_{\mathcal{S}_K}\right) \in \mathbb{R}^{|\mathcal{I}|}. \tag{4}$$

Reducing input sequence length from L to K lowers online inference overhead across paradigms: for self-attention backbones (SASRec, BERT4Rec), quadratic layer costs decrease from $\mathcal{O}(L^2 d)$ to $\mathcal{O}(K^2 d)$, yielding a compute reduction factor of $(K/L)^2$; for state-space architectures (Mamba), compute decreases linearly by a factor of K/L.

3.3 Choice of XAI Methods

We consider four post-hoc XAI methods that are compatible with transformer-based sequential recommendation architectures. **Integrated Gradients (IG)** (Sundararajan et al., 2017) attributes predictions by integrating gradients from a zero baseline to the input. **GradientSHAP** (Lundberg & Lee, 2017) approximates SHAP values via expected gradients over noise-perturbed baselines. **Attention tracing** (Wu et al., 2020) propagates attention weights across layers to obtain a single attribution per input position without requiring an additional backward pass. **Grad-SAM** (Attention$\times$Gradient) (Barkan et al., 2021) combines attention weights with their gradients to highlight positions that are both structurally attended to and gradient-relevant. These four methods span gradient-based, Shapley-inspired, and attention-based attribution families, providing a diverse evaluation of temporal relevance in sequential recommendation. The implementation details for XAI methods are given in Appendix A.2.

Methods such as LIME (Ribeiro et al., 2016) and DeepLIFT (Shrikumar et al., 2017) were excluded: the former is prohibitively slow at long sequence lengths, and the latter is incompatible with the residual connections present in transformer architectures.

3.4 Metrics to Quantify the Benefit of Sequence Distillation

To evaluate the effect of sequence distillation across all compression levels, we consider different values for K : $K \in [1, L]$ and define two summary metrics based on the area under the distillation curve.

Normalised AUC. For a given method and metric $m \in \{\text{HR@10}, \text{NDCG@10}\}$, we first normalise the metric at each distillation level K by its value at the full sequence length L:

$$\tilde{m}(K) = \frac{m(K)}{m(L)}, \tag{5}$$

so that $\tilde{m}(L) = 1$ by construction. We then compute the area under the curve $\tilde{m}(K)$ over all evaluated K values via the trapezoidal rule, normalised by the range of K:

$$\text{AUC}_m = \frac{1}{L-1} \int_1^L \tilde{m}(K) \mathrm{d}K. \tag{6}$$

A value of $\text{AUC}_m > 1$ indicates that, at some distilled sequence length, the model performs better than on the full sequence ($K = L$), while $\text{AUC}_m < 1$ indicates the performance degraded after

distillation. Overall, a higher AUC_m for one method relative to another indicates that it performs better after distillation. This metric captures the overall quality of a distillation method across the entire compression spectrum rather than at a single operating point.

Performance–efficiency trade-off. To characterize the optimal efficiency potential across distillation methods, we complement the holistic AUC_m metric with a 2D Pareto scatter plot mapping peak normalized performance against compute. Specifically, we plot normalized m against normalized GFLOPs at the empirical peak distillation length $K^* = \arg\max_{K<L} \mathrm{m}(K)$ using the coordinate tuple $\left(\frac{\mathrm{GFLOPs}(K^*)}{\mathrm{GFLOPs}(L)}, \frac{\mathrm{m}(K^*)}{\mathrm{m}(L)}\right)$. This visualization maps the upper bound of the quality–efficiency Pareto frontier across methods. Methods situated in the **upper-left region** are optimal, demonstrating that high target performance can be recovered at a fraction of full-sequence inference cost. In production environments where K is fixed offline via a target latency budget or validation set $\mathcal{D}_{\text{valid}}$, this frontier establishes the achievable performance ceiling for a given computational constraint. It is important to note that the reported GFLOPs figures include the cost of projecting the final hidden state onto the full item catalog (the O($|\mathcal{I}|$·d) scoring step).

3.5 Datasets

We evaluate our approach on three public benchmarks: KuaiRand-1k (Gao et al., 2022), MovieLens-1M, and MovieLens-10M (Harper & Konstan, 2015). Each dataset is partitioned into $\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{valid}}$, and $\mathcal{D}_{\text{test}}$, with all evaluation performed on $\mathcal{D}_{\text{test}}$ under the next-item prediction task (ranking the full item vocabulary given an interaction sequence). KuaiRand-1k is a short-video interaction dataset evaluated at two sequence lengths, $L = 500$ and $L = 1000$ (KuaiRand-1k-LONG), to analyze explainability behavior over extended interaction horizons. MovieLens-1M and MovieLens-10M are movie rating benchmarks evaluated at $L = 200$ and $L = 500$, respectively, following standard conventions (Kang & McAuley, 2018; Sun et al., 2019). Table 2 summarizes the dataset statistics.

3.6 Baseline Methods

We compare XAI-based distillation against two model-agnostic baselines that select retained subsequences without attribution signals. **Random distillation** selects K positions uniformly at random as a reference baseline. **Recency-based distillation** retains the K most recent interactions, representing the standard heuristic that recent user history dominates next-item prediction. Both baselines are evaluated under identical settings to isolate the specific impact of attribution-guided selection.

To evaluate generalization across architectural paradigms, we conduct our study across four distinct sequence backbones: **SASRec** (Kang & McAuley, 2018) (autoregressive Transformer), **BERT4Rec** (Sun et al., 2019) (bidirectional Transformer), **Mamba** (Gu & Dao, 2023) (state-space model), and **LONGER** (Chai et al., 2025) (linear-complexity architecture). All distillation strategies are evaluated across these architectures under an identical protocol. For non-attention models (Mamba and LONGER), Attention Tracing and Grad-SAM are excluded as they rely on explicit attention matrices; Integrated Gradients and GradientSHAP are evaluated across all four architectures.

3.7 Evaluation Strategy

We evaluate all distillation methods by measuring downstream ranking performance as a function of the retained sequence length K. For MovieLens(ML)-1M ($L = 200$), we sweep $K \in \{200, 175, 150, 125, 100, 50, 25, 12, 6, 3, 1\}$. For ML-10M ($L = 500$) and KuaiRand-1k ($L = 500$), we sweep $K \in \{500, 400, 300, 200, 100, 75, 50, 25, 12, 6, 3, 1\}$, and for KuaiRand-1k-Long ($L = 1000$), we extend this sweep to $K \in \{1000, 900, 800, 700, 600, 500, 400, 300, 200, 100, 75, 50, 25, 12, 6, 3, 1\}$ to probe distillation behaviour over a longer interaction horizon. In every case, $K = L$ corresponds to the full, undistilled sequence and serves as the reference point for the normalized metrics described in Equation 5.

All models are trained with PyTorch using the Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.98$) with a learning rate of 1e-3. For each backbone/attention/dataset configuration, we train and evaluate across 3 random seeds and report the mean and standard deviation of all metrics across seeds, ensuring that reported differences between distillation methods are not artifacts of a single training run. More training/model details are given in Section A.1 in the Appendix.

4 RESULTS AND DISCUSSIONS

We organize our findings around three questions: does XAI-driven distillation improve ranking quality over attribution-agnostic baselines across the full compression spectrum (Section 4.1); does this advantage translate into a favorable performance-efficiency trade-off at each method's best operating point (Section 4.2); and does the benefit of attribution-guided distillation persist under varying model capacity and noisy input sequences (Section 4.3).

Table 1: $AUC_{NDCG@10}$ (mean ± std across seeds) across different model architectures.

Method	Submethod	ML-1M	ML-10M	KuaiRand-1K	KuaiRand-1K-Long
SASRec					
Random	–	0.68 ± 0.013	0.72 ± 0.060	0.90 ± 0.065	0.94 ± 0.083
Recency	–	**0.99 ± 0.028**	0.98 ± 0.056	1.01 ± 0.13	1.04 ± 0.13
Ours (XAI-Driven)	Integrated Gradient	0.97 ± 0.022	0.97 ± 0.047	1.39 ± 0.33	1.15 ± 0.052
	GradientSHAP	0.95 ± 0.016	**0.99 ± 0.058**	**1.47 ± 0.28**	**1.21 ± 0.022**
	Attention Tracing	0.80 ± 0.029	0.80 ± 0.043	1.30 ± 0.49	1.06 ± 0.091
	Grad-SAM	0.93 ± 0.027	0.88 ± 0.038	1.41 ± 0.44	1.10 ± 0.086
BERT4Rec					
Random	–	0.57 ± 0.030	0.69 ± 0.025	1.05 ± 0.076	1.02 ± 0.065
Recency	–	**0.87 ± 0.062**	0.90 ± 0.025	0.95 ± 0.057	1.02 ± 0.046
Ours (XAI-Driven)	Integrated Gradient	0.85 ± 0.061	0.91 ± 0.014	1.36 ± 0.41	**1.25 ± 0.24**
	GradientSHAP	0.82 ± 0.067	0.90 ± 0.024	1.43 ± 0.43	1.14 ± 0.19
	Attention Tracing	0.86 ± 0.046	**0.94 ± 0.014**	0.82 ± 0.15	1.17 ± 0.15
	Grad-SAM	0.80 ± 0.034	0.83 ± 0.027	**1.59 ± 0.94**	1.11 ± 0.082
Mamba					
Random	–	0.68 ± 0.035	0.71 ± 0.022	0.82 ± 0.12	0.90 ± 0.082
Recency	–	**0.99 ± 0.016**	**0.97 ± 0.032**	0.84 ± 0.042	0.98 ± 0.10
Ours (XAI-Driven)	Integrated Gradient	0.96 ± 0.024	0.96 ± 0.014	**1.04 ± 0.069**	**1.11 ± 0.13**
	GradientSHAP	0.95 ± 0.030	0.96 ± 0.036	1.03 ± 0.079	1.05 ± 0.12
LONGER					
Random	–	0.58 ± 0.022	0.58 ± 0.059	0.80 ± 0.11	0.83 ± 0.055
Recency	–	**0.85 ± 0.038**	**0.82 ± 0.083**	0.82 ± 0.21	0.94 ± 0.010
Ours (XAI-Driven)	Integrated Gradient	0.79 ± 0.051	0.80 ± 0.042	1.03 ± 0.12	**1.20 ± 0.093**
	GradientSHAP	0.77 ± 0.059	0.79 ± 0.10	**1.08 ± 0.12**	1.10 ± 0.095
	Attention Tracing	0.71 ± 0.038	0.71 ± 0.050	0.91 ± 0.18	0.71 ± 0.099
	Grad-SAM	0.76 ± 0.029	0.75 ± 0.055	0.99 ± 0.15	0.86 ± 0.028

4.1 EFFECTIVENESS OF XAI-DRIVEN SEQUENCE DISTILLATION

Table 1 reports $\mathrm{AUC}_{\mathrm{NDCG@10}}$, as defined in Equation 6 and computed from the curves in Figure 6 in the Appendix. This summary statistic captures how well a method preserves ranking quality across the entire compression spectrum rather than at any single retained length K. Aggregated across all backbone architectures (SASRec, BERT4Rec, Mamba, LONGER) and datasets (ML-1M, ML-10M, KuaiRand-1K, KuaiRand-1K-Long), XAI-driven distillation delivers a consistent net advantage over the Recency baseline. Averaged over all evaluated settings, Integrated Gradients and GradientSHAP improve over Recency's mean $\mathrm{AUC}_{\mathrm{NDCG@10}}$ by $+11.5\%$ and $+11.0\%$, respectively, while Grad-SAM improves by $+6.9\%$. Attention Tracing is the exception, trailing Recency by 3.9% on average. This indicates that raw attention weights are not a reliable indicator of sequence position importance, a limitation consistent with prior literature (Chefer et al., 2021; Serrano & Smith, 2019; Jain & Wallace, 2019; Wiegreffe & Pinter, 2019; Pandey et al., 2026). This pattern is corroborated across dataset-architecture combinations in Figure 2, where Integrated Gradients and GradientSHAP display the most consistent improvements over *Random* method while Attention Tracing is the only method whose performance distribution dips into negative territory. Crucially, this performance edge is directional and consistent across all settings: Integrated Gradients and GradientSHAP exceed Recency in all eight KuaiRand configurations (4 backbones × 2 sequence lengths) in Table 1. This strict sign-consistency demonstrates that the relative advantage is robust to per-cell variance.

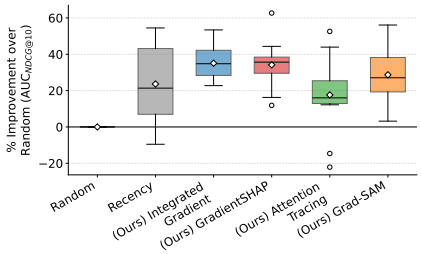


Figure 2: Percentage improvement in $AUC_{NDCG@10}$ over the **Random** baseline.

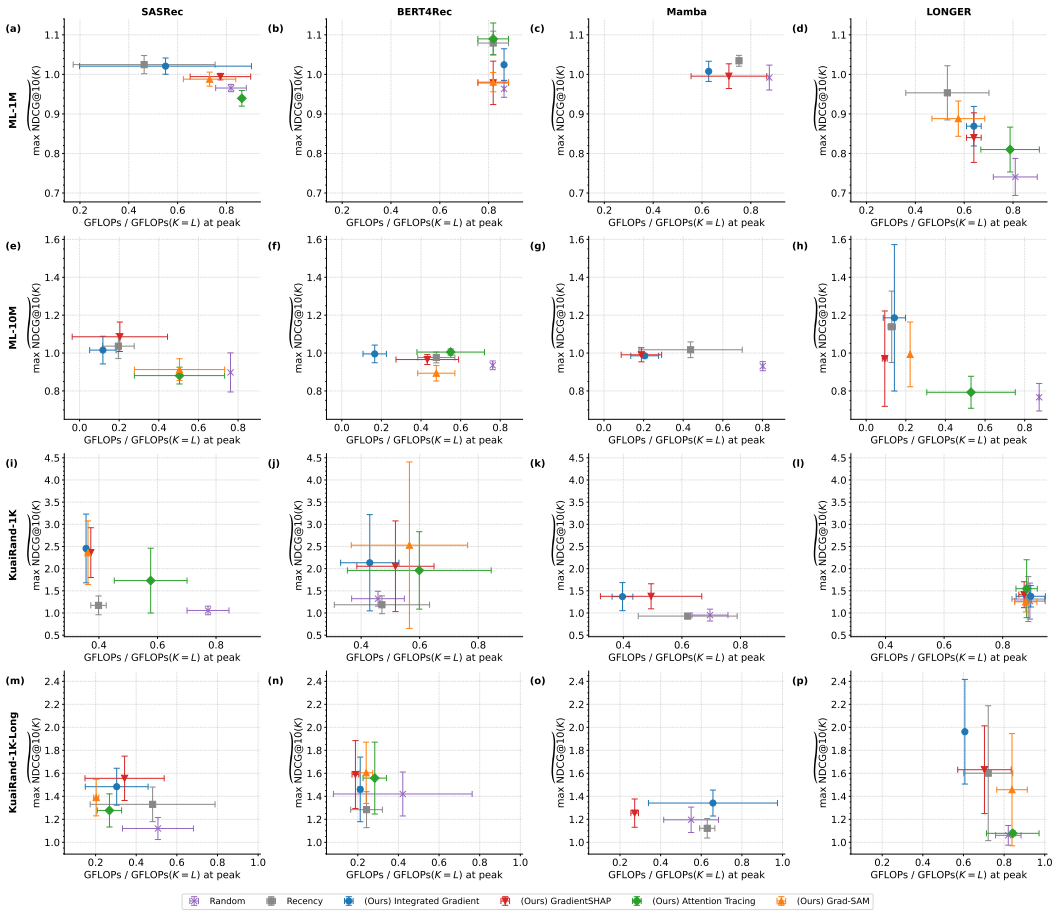


Figure 3: **Peak normalised NDCG@10 vs. normalised GFLOPs** across model architectures (columns) and datasets (rows). Points closer to the upper-left (higher normalised NDCG@10 at lower normalised GFLOPs) represent a **better performance-efficiency trade-off**.

This overall advantage varies across datasets: on ML-1M and ML-10M, Recency outperforms XAI-driven methods on average across backbones, with a narrower gap on ML-10M (-1.0% to -9.2%) than on ML-1M (-3.6% to -12.6%). We attribute this to strong recency bias inherent to MovieLens benchmarks, where a user's next choice is disproportionately predictable from recent interactions (Figure 11). Under such dynamics, retaining the sequence tail serves as a near-optimal heuristic, leaving limited margin for attribution-based gains. Although Recency leads on MovieLens in mean $\mathrm{AUC}_{\mathrm{NDCG@10}}$, this edge disappears when evaluated under practical lenses: performance-efficiency trade-offs and noise robustness, where XAI-guided distillation matches or exceeds Recency. Figure 12 further shows this behavior is independent of attention architecture: across standard, linear, and sparse attention SASRec on ML-10M and KuaiRand-1K, Integrated Gradients yields near-identical performance-GFLOPs curves across attention variants. This confirms sequence distillation generalizes across attention mechanisms, driven primarily by dataset-level temporal dynamics. Conversely, KuaiRand-1K exhibits far less recency bias due to distributed short-video engagement signals. Here, attribution-based selection achieves its largest gains, with Integrated Gradients, GradientSHAP, and Grad-SAM exceeding Recency by over 30% in mean AUC. This advantage persists on KuaiRand-1K-Long ($+2\%$ to $+19\%$), proving the benefits stem from underlying engagement dynamics rather than sequence length alone. Corresponding $\mathrm{AUC}_{\mathrm{HR@10}}$ results appear in Table 4 in the Appendix.

4.2 XAI-driven Sequence Distillation offer Higher Efficiency

While $\mathrm{AUC}_{\mathrm{NDCG@10}}$ summarizes performance across the full spectrum of compression levels K, it does not indicate the computational savings achieved at each method's optimal operating point. Figure 3 complements Table 1 by plotting $\max_K \widetilde{\mathrm{NDCG@10}}(K)$ (as defined in Equation 5) against

the corresponding normalized GFLOPs, $\text{GFLOPs}(K)/\text{GFLOPs}(L)$, at that peak K for SASRec across all four datasets (panels a, e, i, m). Points with higher $\max_K \widetilde{\text{NDCG@10}}(K)$ and lower $\text{GFLOPs}(K)/\text{GFLOPs}(L)$ are preferred, as they indicate a method that recovers or exceeds full-sequence ranking quality at a fraction of the inference cost.

This perspective demonstrates that XAI-driven methods are not only competitive on AUC but strictly Pareto-dominate Recency at their peak operating points on three of the four datasets, achieving higher performance with lower compute. On KuaiRand-1K (panel i), Integrated Gradients and GradientSHAP reach $\max_K \widetilde{\text{NDCG@10}}(K) = 2.46$ and 2.36 using only 36% and 37% of $\text{GFLOPs}(L)$, respectively, whereas Recency requires higher compute (40% of $\text{GFLOPs}(L)$) to reach a substantially lower peak of $\widetilde{\text{NDCG@10}} = 1.17$. A similar pattern holds on KuaiRand-1K-Long (panel m): GradientSHAP and Integrated Gradients achieve 1.56 and 1.48 at 34% and 30% of $\text{GFLOPs}(L)$, while Recency reaches a weaker peak of 1.33 at 48% compute. On ML-10M (panel e), GradientSHAP outperforms Recency's peak (1.09 vs. 1.04) at an equivalent compute budget (20%), while Integrated Gradients matches Recency's peak within 2% (1.02 vs. 1.04) while requiring well under half the compute (12% vs. 20%).

The single exception is ML-1M (panel a), where Recency reaches its peak (1.025 at 46% of $\text{GFLOPs}(L)$) at lower compute than Integrated Gradients' near-identical peak (1.021 at 55%). Even here, the performance gap is under half a percent, contrasting with the $AUC_{NDCG@10}$ gap reported in Table 1. This confirms that Recency's AUC_m advantage on MovieLens is concentrated away from the optimal operating point. When compared at their respective peaks, explainability-guided distillation effectively ties with Recency on the single dataset where it does not outright dominate, while strictly outperforming it across all other benchmarks. Overall, these results show that explainability-guided distillation Pareto-dominates or closely tracks recency truncation across all datasets for SASRec, realizing the largest efficiency gains when temporal signal is not concentrated strictly in the most recent interactions. The remaining architecture columns for BERT4Rec, Mamba, and LONGER in Figure 3 demonstrate identical efficiency trends for $\widetilde{\text{NDCG@10}}$, while Figure 8 in the Appendix shows similarly consistent patterns for $\widetilde{\text{HR@10}}$ across all backbones. Although offline attributions incur a periodic precomputation cost, our framework strictly optimizes online serving efficiency. By pruning user sequences offline, online (live) request processing achieves reduction in serving GFLOPs as discussed above.

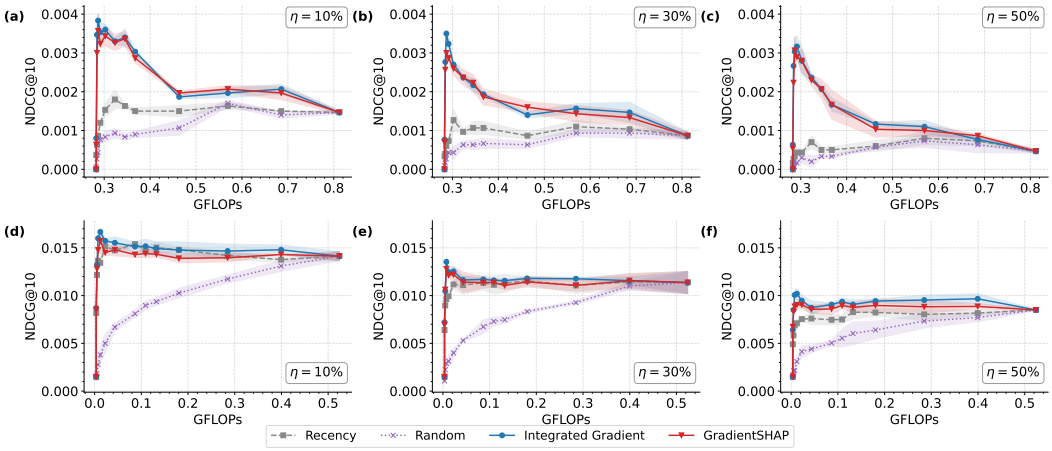


Figure 4: NDCG@10 as a function of GFLOPs for SASRec on KuaiRand-1K (a-c) and ML-1M (d-f) under random item noise injection at three noise levels ($\eta \in \{10\%, 30\%, 50\%\}$). The **rightmost point** in each graph represents the performance with **no distillation**.

4.3 Ablation Study

4.3.1 Impact of XAI-driven sequence distillation for noisy input

To assess whether attribution-guided distillation remains beneficial when the interaction history is corrupted, we inject random noise into test sequences before computing attributions and evaluating downstream performance. For a noise level $\eta \in \{10\%, 30\%, 50\%\}$, we independently sample $\lceil \eta \cdot L \rceil$ positions per sequence without replacement and replace each sampled item with a uniform sample

from the full vocabulary. This corruption is applied exclusively to $\mathcal{D}_{\text{test}}$, leaving training and validation sequences unperturbed. Attribution scores are then recomputed on each corrupted sequence to guide distillation to varying target lengths K, following the protocol in Section 3.2.

Figure 4 reports NDCG@10 against GFLOPs for SASRec under this setup, with KuaiRand-1K in the top row (panels a–c) and ML-10M in the bottom row (panels d–f). On KuaiRand-1K, XAI-guided distillation provides a substantially larger benefit than Recency across all noise levels: Integrated Gradients and GradientSHAP sustain a peak between 0.003 and 0.004 NDCG@10, while Recency remains well below this plateau. On ML-10M, while clean-input benchmarks show comparable performance between Recency and attribution-based methods (Table 1, Figure 3), XAI-guided methods demonstrate superior resilience as sequence noise increases. Crucially, across both datasets, explainability-driven distillation improves performance over the uncompressed sequence ($K = L$), showcasing a clear denoising effect that filters out corrupted interactions. Averaged across all noise levels and both datasets, Integrated Gradients and GradientSHAP outperform Recency by 37% and 33%, respectively, in $\text{AUC}_{\text{NDCG@10}}$, demonstrating that attribution-guided selection remains the more reliable distillation strategy under input noise.

To simulate real-world noise from accidental clicks or transient exploration, we corrupt the most recent interactions in test sequences (Figure 13). On KuaiRand-1K, Integrated Gradients and GradientSHAP remain robust across all noise levels $\eta \in \{10\%, 30\%, 50\%\}$, whereas recency truncation degrades sharply below random selection. On ML-10M, while all methods decline due to strong temporal reliance (Figure 11), attribution-guided methods widen their relative advantage. Across both datasets and noise ratios, Integrated Gradients and GradientSHAP achieve nearly twice the $\text{AUC}_{\text{NDCG@10}}$ of Recency, demonstrating that XAI-guided distillation provides critical resilience when recent interactions are unreliable.

4.3.2 Effect of model size on sequence distillation

Figure 5 characterizes the interaction between model capacity and XAI-guided distillation for SASRec on KuaiRand-1K, sweeping four parameter configurations and reporting raw NDCG@10 as a function of the distilled sequence length K. Capacity governs both full-sequence performance and the achievable peak after distillation: the two largest models (138K and 341K) dominate throughout the compression spectrum, attaining both the highest full-sequence NDCG@10 and the highest peaks once Integrated Gradients selects a small, high-value subset of positions. The compact models (29K and 61K parameters) lag behind at all sequence lengths K, with the performance gap expanding under higher compression. This behavior stems from under-parameterization, where restricted model capacity limits representations learned from the full history. Although distillation provides relative gains over the undistilled baseline, its absolute ceiling remains bounded by the model's base capacity. Thus, XAI-guided distillation relies on sufficient architectural capacity to maximize efficiency gains.

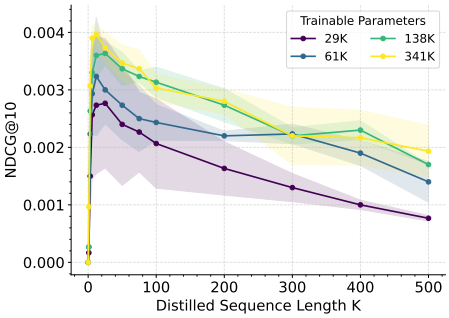


Figure 5: NDCG@10 as a function of distilled sequence length K for SASRec on KuaiRand-1K, using Integrated Gradients across four model sizes. Shaded regions denote standard deviation across three seeds.

5 Conclusion

We present explainability-guided sequence distillation, a retraining-free framework leveraging post-hoc attribution to distill user interaction histories and drastically reduce online inference GFLOPs. Across four backbone architectures and four dataset configurations, our approach consistently outperforms standard recency truncation, matching or exceeding it even under heavy recency bias when evaluating budget-matched compute. Comprehensive ablations show these gains persist and amplify under sequence corruption. Ultimately, our work repurposes explainability from a passive diagnostic into an active driver of production online inference efficiency across diverse sequential recommendation models. Future directions include live deployment and extension to broader sequential domains.

AI USE STATEMENT

In this work, we used generative AI tools for refining the textual draft and generating visualization code for figures. We take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our findings, we have provided comprehensive details of our experimental setup, architectural parameters, and evaluation metrics in Section 3 and Appendix A. Furthermore, the complete codebase, including data preprocessing scripts, backbone model implementations, post-hoc attribution pipelines, and evaluation routines, is included as supplementary material and will be made fully open-source upon paper acceptance.

REFERENCES

Oren Barkan, Edan Hauon, Avi Caciularu, Ori Katz, Itzik Malkiel, Omri Armstrong, and Noam Koenigstein. Grad-sam: Explaining transformers via gradient self-attention maps. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 2882–2887, 2021.

Zheng Chai, Qin Ren, Xijun Xiao, Huizhi Yang, Bo Han, Sijun Zhang, Di Chen, Hui Lu, Wenlin Zhao, Lele Yu, et al. Longer: Scaling up long sequence modeling in industrial recommenders. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, pp. 247–256, 2025.

Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 782–791, 2021.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Chongming Gao, Shijun Li, Yuan Zhang, Jiawei Chen, Biao Li, Wenqiang Lei, Peng Jiang, and Xiangnan He. Kuairand: An unbiased sequential recommendation dataset with randomly exposed videos. In *Proceedings of the 31st ACM international conference on information & knowledge management*, pp. 3953–3957, 2022.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4), December 2015. ISSN 2160-6455. doi: 10.1145/2827872.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. *CoRR*, abs/2111.06377, 2021. URL https://arxiv.org/abs/2111.06377.

Sarthak Jain and Byron C. Wallace. Attention is not Explanation. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3543–3556, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1357. URL https://aclanthology.org/N19-1357/.

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

Dwipam Katariya, Snehita Varma, Akshat Shreemali, Benjamin Wu, Kalanand Mishra, and Pranab Mohanty. Fintrec: Transformer based unified contextual ads targeting and personalization for financial applications. *arXiv preprint arXiv:2511.14865*, 2025.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pp. 5156–5165. PMLR, 2020.

Jiayi Liao, Ruobing Xie, Sihang Li, Xiang Wang, Xingwu Sun, Zhanhui Kang, and Xiangnan He. Multi-grained patch training for efficient llm-based recommendation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, pp. 1572–1581, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400715921. doi: 10.1145/3726302.3730042. URL https://doi.org/10.1145/3726302.3730042.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

Akash Pandey, Kanisha Shah, Addrish Roy, Dwipam Katariya, Hongyangyang Shi, Amanda Ding, Kalanand Mishra, and Pranab Mohanty. A systematic benchmark of explainable methods for temporal attribution in sequential recommendation systems, 2026. URL https://arxiv.org/abs/2609.27201.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=kR95DuwwXHZ.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

Sofia Serrano and Noah A Smith. Is attention interpretable? *arXiv preprint arXiv:1906.03731*, 2019.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pp. 3145–3153. PMlR, 2017.

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*, pp. 1441–1450, 2019.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319–3328. PMLR, 2017.

Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 11–20, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1002. URL https://aclanthology.org/D19-1002/.

Zhengxuan Wu, Thanh-Son Nguyen, and Desmond C Ong. Structured self-attention weights encode semantics in sentiment analysis. *arXiv preprint arXiv:2010.04922*, 2020.

Table 2: Dataset statistics.

Dataset	L	**Users**	**Items**	**Sparsity**	$\lvert\mathcal{D}_{train}\rvert$	$\lvert\mathcal{D}_{valid}\rvert$	$\lvert\mathcal{D}_{test}\rvert$
KuaiRand-1K	500	922	1,101,222	99.75%	28,267	6,149	6,553
KuaiRand-1K-LONG	1000	786	1,068,905	99.72%	22,314	4,858	5,205
MovieLens-1M	200	1,578	3,883,	89.31%	17,731	2,813	3,080
MovieLens-10M	500	3,659	10,583	91.99%	17,973	4,672	4,728

A Model Details

A.1 Training Details

All models were trained using NVIDIA A10G GPUs for a total of 200 epochs. Table 3 details the number of trainable parameters for the SASRec and BERT4Rec architectures across the KuaiRand datasets.

Table 3: Number of trainable parameters for SASRec, BERT4Rec, Mamba, and LONGER architectures across datasets.

Architecture	**Variant**	**KuaiRand-1k**	**KuaiRand-1k-LONG**	**MovieLens-1M**	**MovieLens-10M**
Transformer	SASRec	341K	405K	1.56M	2.15M
	BERT4Rec	341K	405K	1.56M	2.15M
Non-Transformer	Mamba	442K	506K	1.55M	2.14M
	LONGER	524K	588K	1.74M	2.34M

A.2 Integrated Gradient and GradientSHAP hyperparameters:

For Integrated Gradients, we integrate over $n_steps = 20$ interpolation steps from a zero baseline (i.e., the all-zero item-embedding tensor), matching Captum's default Riemann-sum approximation. For GradientSHAP, we draw $n_samples = 2$ perturbed baselines per input from a Gaussian noise distribution centered at zero with standard deviation 0.001 (four such noise baselines are pre-sampled and shared across the batch), and apply a smoothing noise level of $stdevs = 0.05$ around each sampled baseline, following Captum's implementation of the expected-gradients formulation. Both methods attribute importance with respect to the single top-predicted next item ($top_k = 1$) at each position, rather than aggregating over multiple candidate targets. These settings are held fixed across all architectures (SASRec, BERT4Rec, Mamba, LONGER), attention variants, and datasets to ensure a controlled comparison; we did not perform hyperparameter tuning for either attribution method, as our goal is to evaluate off-the-shelf explainability signals rather than to optimize attribution quality itself.

B ADDITIONAL RESULTS

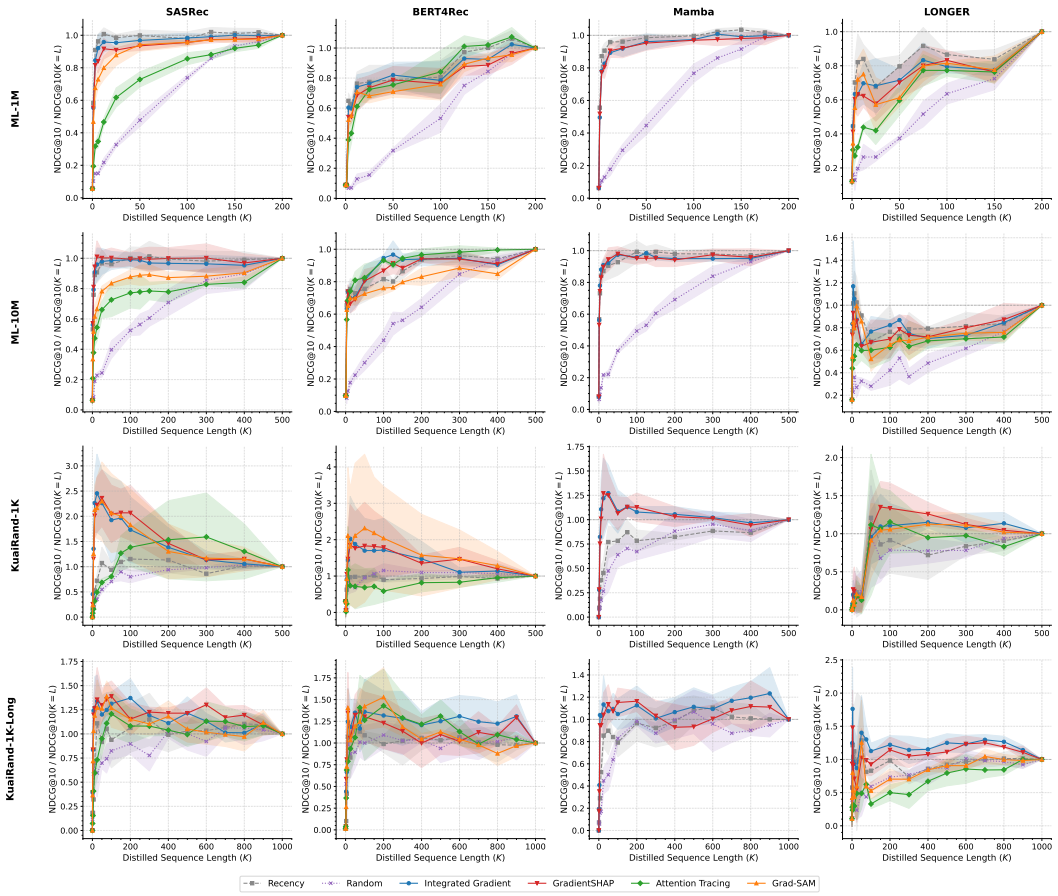


Figure 6: **Normalised NDCG@10 vs. distilled sequence length** K for SASRec, BERT4Rec, Mamba, and LONGER (columns, left to right; standard attention only where applicable) evaluated on ML-1M, ML-10M, KuaiRand-1K, and KuaiRand-1K-Long (rows, top to bottom). Each curve shows NDCG@10(K)/NDCG@10(K=L), i.e. performance at truncation length K normalised by the full-sequence (K=L) score, averaged across seeds; shaded regions denote ± 1 standard deviation across seeds. Mamba and LONGER have no explicit attention matrix, so Attention Tracing and Attention $\times$ Gradient are omitted for those columns.

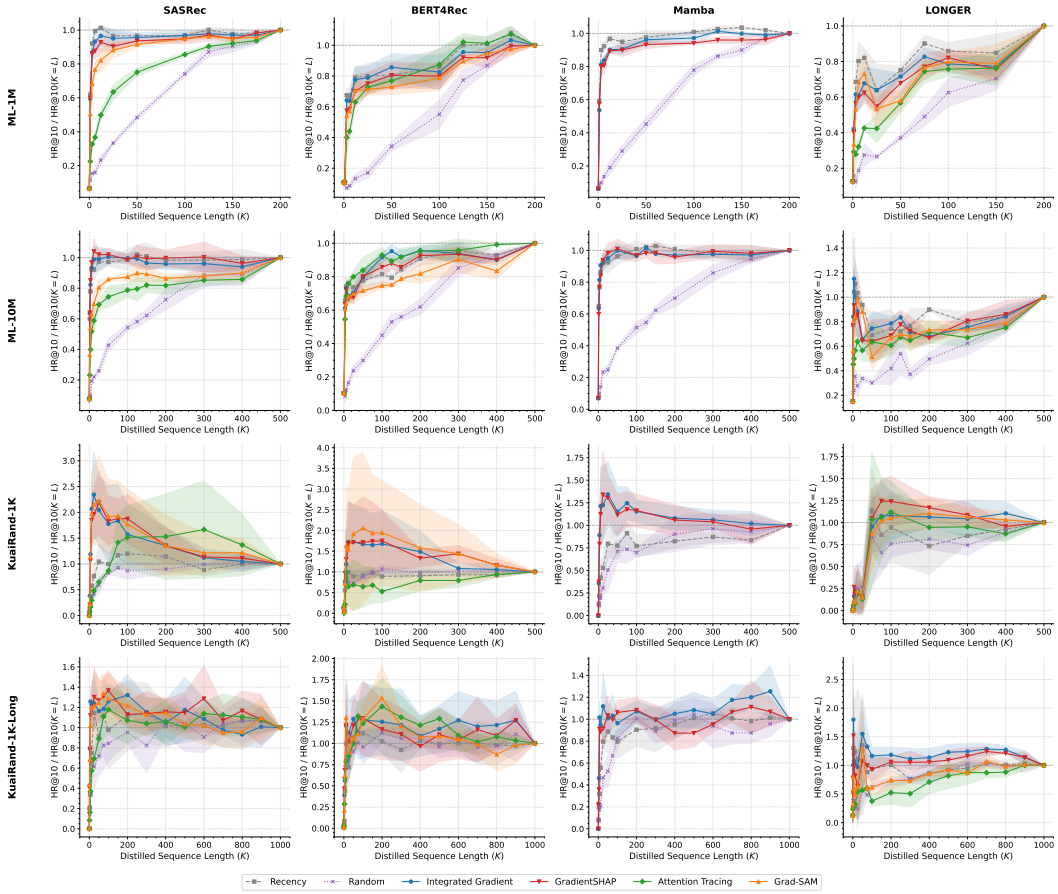


Figure 7: **Normalised HR@10 vs. distilled sequence length** K for SASRec, BERT4Rec, Mamba, and LONGER (columns, left to right; standard attention only where applicable) evaluated on ML-1M, ML-10M, KuaiRand-1K, and KuaiRand-1K-Long (rows, top to bottom). Each curve shows HR@10(K)/HR@10$(K{=}L)$, i.e. performance at truncation length K normalised by the full-sequence ($K{=}L$) score, averaged across seeds; shaded regions denote ± 1 standard deviation across seeds. Mamba and LONGER have no explicit attention matrix, so Attention Tracing and Attention $\times$ Gradient are omitted for those columns.

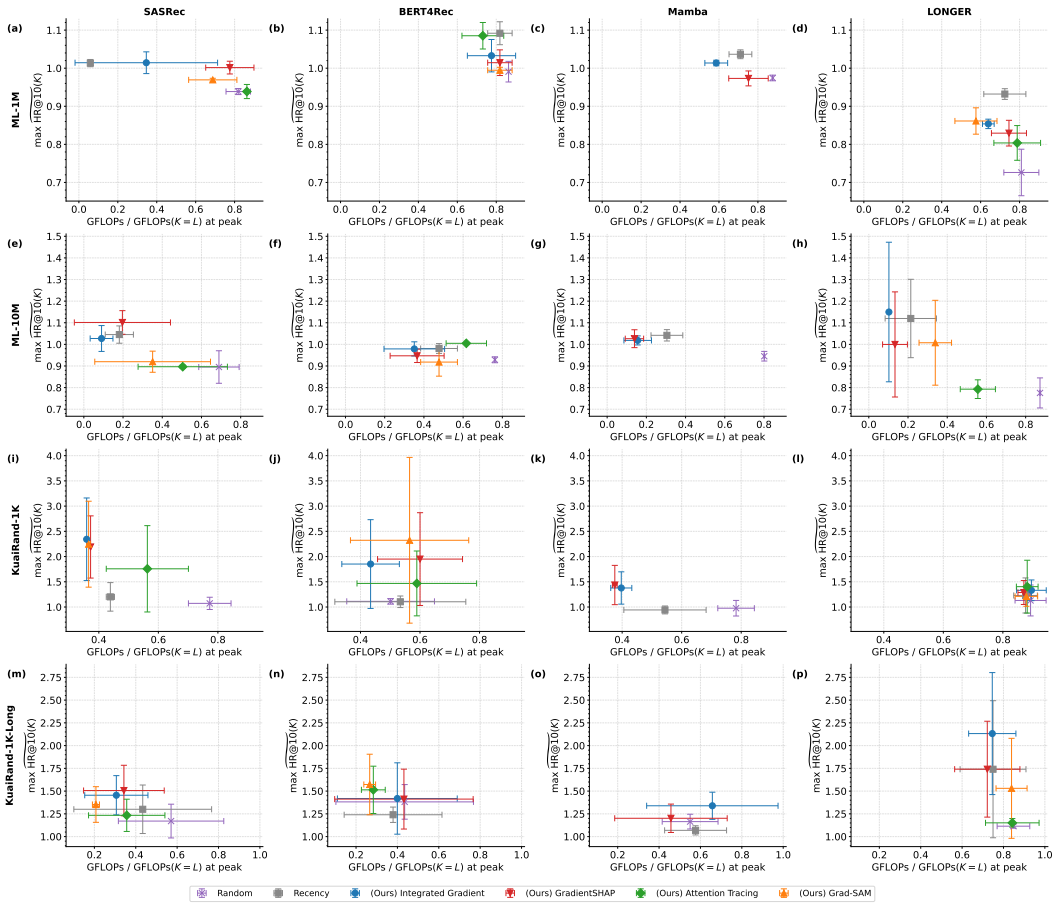


Figure 8: **Peak normalised HR@10 vs. normalised GFLOPs** for SASRec (a, e, i, m), BERT4Rec (b, f, j, n), Mamba (c, g, k, o), and LONGER (d, h, l, p), each with standard attention, evaluated on MovieLens-1M (a–d), ML-10M (e–h), KuaiRand-1K (i–l), and KuaiRand-1K-Long (m–p). Points closer to the **upper-left corner** (higher normalised NDCG@10 at lower normalised GFLOPs) indicate a **better performance-efficiency trade-off**.

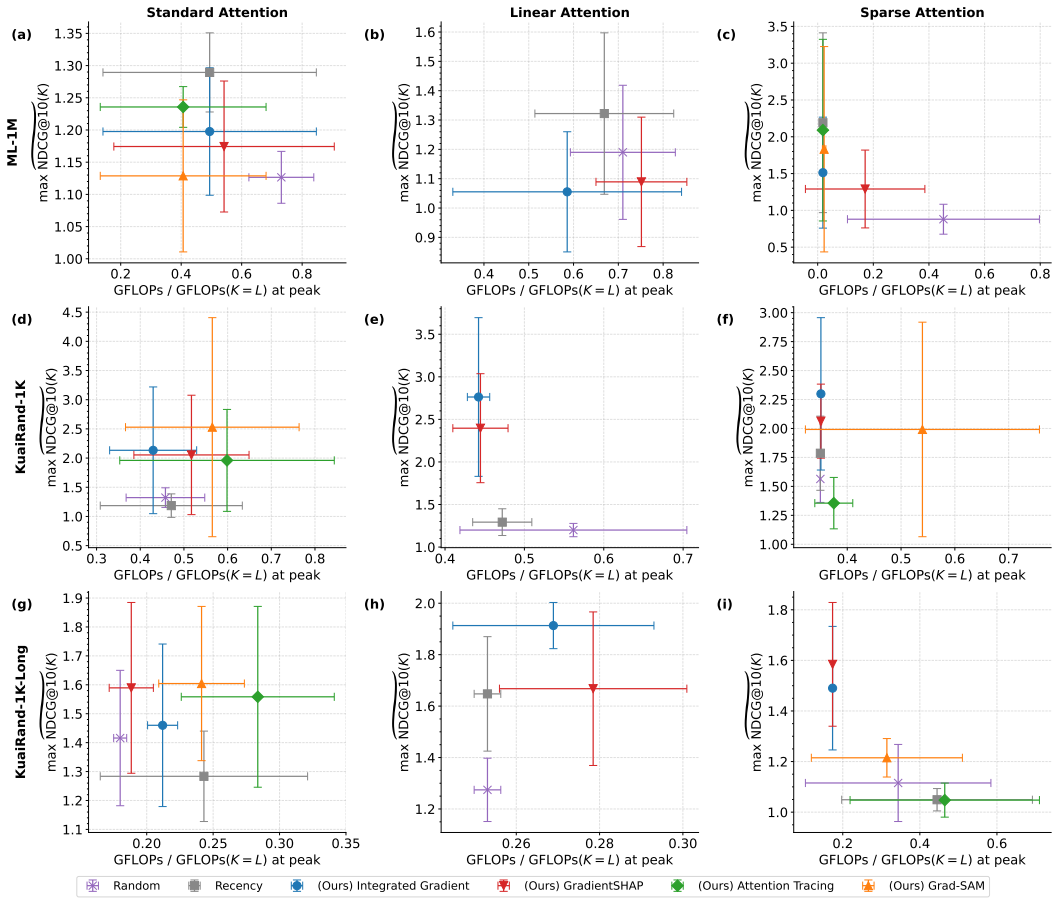


Figure 9: Peak normalised NDCG@10 vs normalised GFLOPs for BERT4Rec on MovieLens-1M with standard attention (a), linear attention (b), and sparse attention (c); on KuaiRand-1K with standard attention (d), linear attention (e), and sparse attention (f); and on KuaiRand-1K-Long with standard attention (g), linear attention (h), and sparse attention (i).

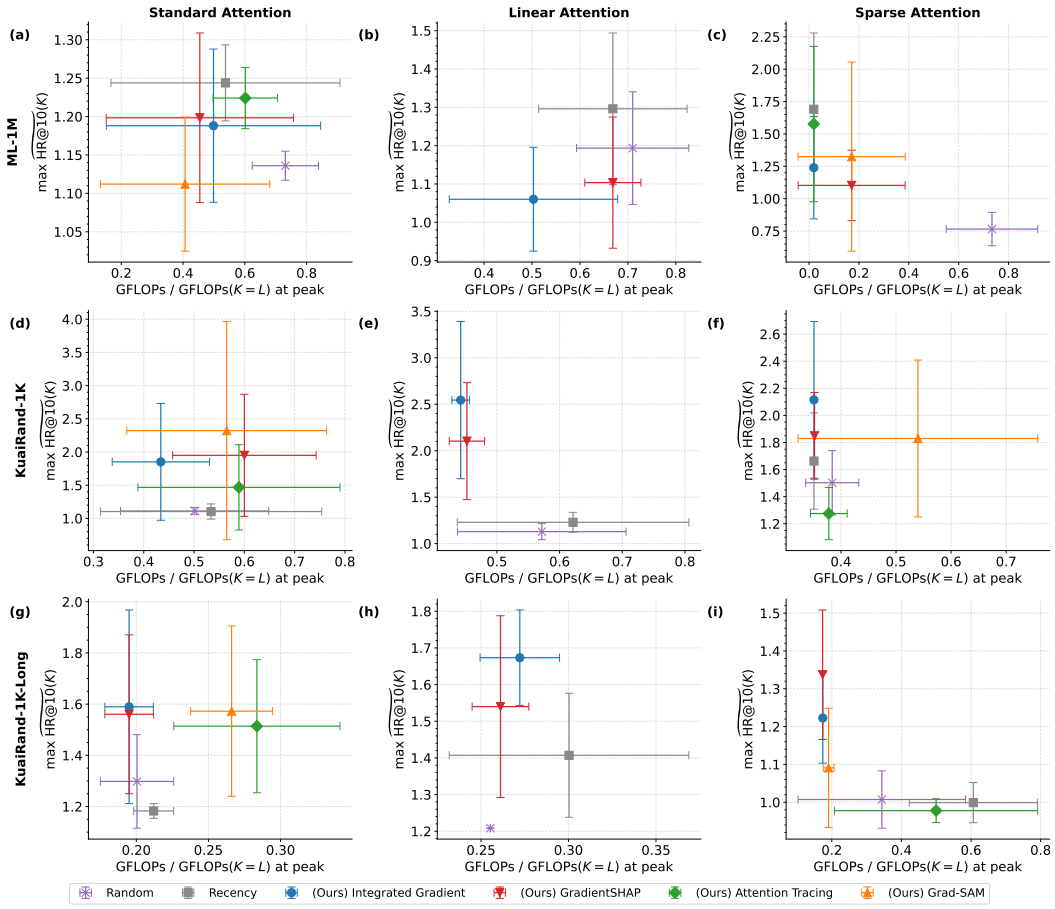


Figure 10: Peak normalised HR@10 vs normalised GFLOPs for BERT4Rec on MovieLens-1M with standard attention (a), linear attention (b), and sparse attention (c); on KuaiRand-1K with standard attention (d), linear attention (e), and sparse attention (f); and on KuaiRand-1K-Long with standard attention (g), linear attention (h), and sparse attention (i).

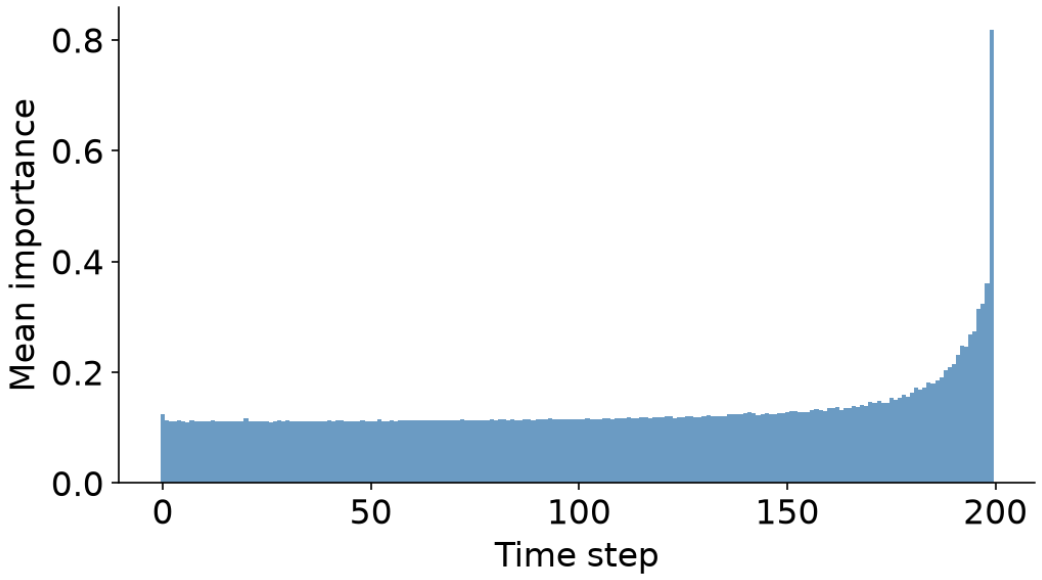


Figure 11: Importance score as function of temporal position, calculated from Grad-SAM method on ML-1M dataset. Higher the temporal position, more recent the point. The plot shows high recency bias.

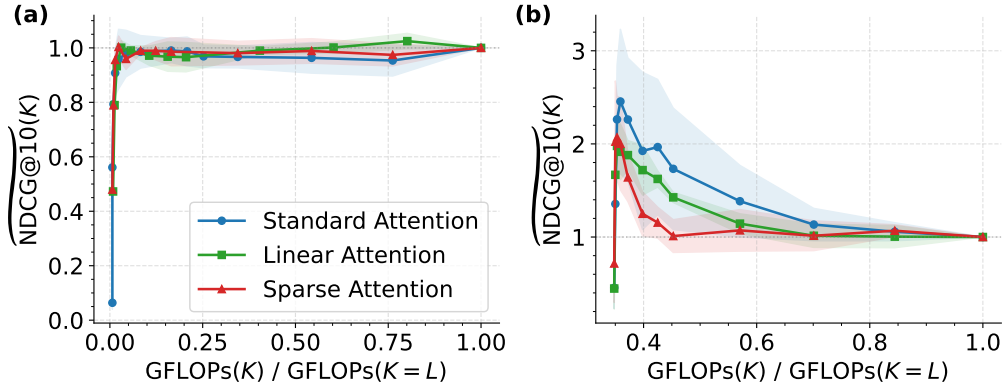


Figure 12: $\widetilde{\text{NDCG@10}}(K)$ vs. normalized GFLOPs for SASRec with Integrated Gradients on ML-10M (a) and KuaiRand-1K (b) across attention variants.

Table 4: $AUC_{HR@10}$ (mean ± std across seeds) across different model architecture.

Method	Submethod	ML-1M	ML-10M	KuaiRand-1K	KuaiRand-1K-Long
SASRec					
Random	–	0.68 ± 0.0094	0.73 ± 0.048	0.91 ± 0.098	0.96 ± 0.15
Recency	–	**0.97 ± 0.011**	0.98 ± 0.049	1.02 ± 0.18	1.04 ± 0.17
Ours (XAI-Driven)	Integrated Gradient	0.96 ± 0.019	0.97 ± 0.040	1.33 ± 0.33	1.10 ± 0.14
	GradientSHAP	0.94 ± 0.017	**0.99 ± 0.048**	1.38 ± 0.24	**1.16 ± 0.15**
	Attention Tracing	0.81 ± 0.0084	0.83 ± 0.025	1.35 ± 0.58	1.06 ± 0.14
	Grad-SAM	0.92 ± 0.016	0.88 ± 0.028	**1.42 ± 0.46**	1.09 ± 0.15
BERT4Rec					
Random	–	0.59 ± 0.037	0.69 ± 0.017	0.99 ± 0.040	1.02 ± 0.066
Recency	–	**0.90 ± 0.061**	0.89 ± 0.019	0.92 ± 0.042	0.99 ± 0.054
Ours (XAI-Driven)	Integrated Gradient	0.88 ± 0.057	0.91 ± 0.015	1.31 ± 0.38	**1.20 ± 0.31**
	GradientSHAP	0.84 ± 0.053	0.89 ± 0.021	1.39 ± 0.36	1.12 ± 0.27
	Attention Tracing	0.87 ± 0.036	**0.93 ± 0.016**	0.79 ± 0.11	1.16 ± 0.12
	Grad-SAM	0.82 ± 0.032	0.83 ± 0.032	**1.50 ± 0.84**	1.10 ± 0.13
Mamba					
Random	–	0.68 ± 0.019	0.72 ± 0.017	0.85 ± 0.13	0.90 ± 0.079
Recency	–	**0.99 ± 0.015**	**0.98 ± 0.028**	0.83 ± 0.052	0.95 ± 0.094
Ours (XAI-Driven)	Integrated Gradient	0.96 ± 0.0050	0.97 ± 0.019	**1.09 ± 0.12**	**1.09 ± 0.13**
	GradientSHAP	0.94 ± 0.015	0.98 ± 0.038	1.06 ± 0.13	1.00 ± 0.10
LONGER					
Random	–	0.57 ± 0.010	0.59 ± 0.059	0.78 ± 0.064	0.86 ± 0.026
Recency	–	**0.83 ± 0.0061**	**0.84 ± 0.065**	0.83 ± 0.19	0.96 ± 0.016
Ours (XAI-Driven)	Integrated Gradient	0.78 ± 0.0082	0.79 ± 0.047	0.99 ± 0.14	**1.19 ± 0.11**
	GradientSHAP	0.76 ± 0.035	0.78 ± 0.081	**1.03 ± 0.11**	1.09 ± 0.11
	Attention Tracing	0.70 ± 0.029	0.71 ± 0.040	0.91 ± 0.15	0.74 ± 0.12
	Grad-SAM	0.75 ± 0.013	0.76 ± 0.032	0.98 ± 0.12	0.88 ± 0.053

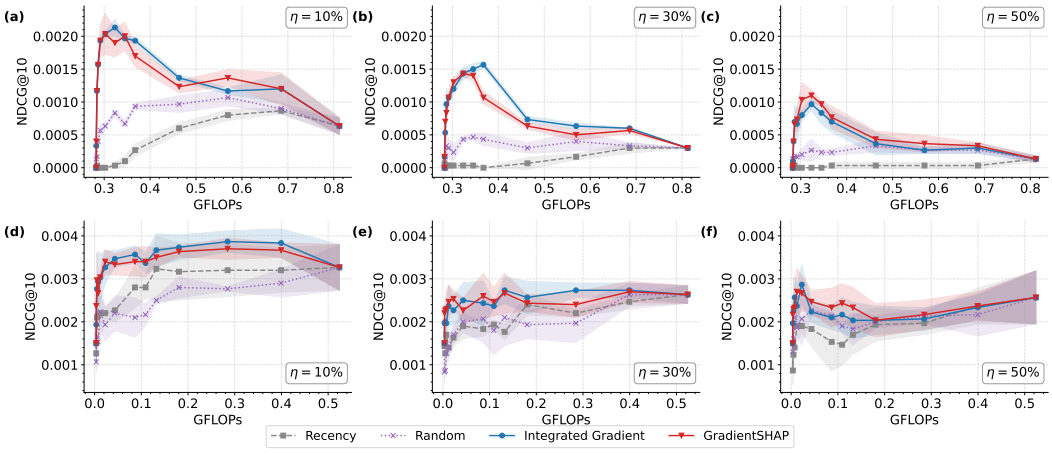


Figure 13: NDCG@10 as a function of GFLOPs for SASRec on KuaiRand-1K (a-c) and ML-1M (d-f) under **recent** item noise injection at three noise levels ($\eta \in \{10\%, 30\%, 50\%\}$). The **rightmost point** in each graph represents the performance with **no distillation**.