Sequential Multimodal Evidence Optimization for Product Media Ranking in E-Commerce

Prasenjit Dey
Amazon.com Inc.
prasendx@amazon.com

Frank McIntyre
Amazon.com Inc.
flm@amazon.com

Arnab Sinha
Amazon.com Inc.
arsinha@amazon.com

ABSTRACT

On modern e-commerce stores, customers consume ordered slates of heterogeneous product media—such as images, videos, and 3D renders—before making purchase decisions. Existing media-ranking systems often optimize myopic engagement proxies such as clicks or dwell time, even though product media assets are cooperative informational components of the same item that together help customers find the information they need through sequential interaction. We present Sequential Multimodal Evidence Optimization (SMEO), a two-stage utility-guided framework for customer-oriented media sequencing. SMEO first learns a trajectory utility model from consumed media prefixes to estimate how ordered evidence helps customers reach a purchase decision, while mitigating position-bias and variable-depth imbalance in logged data. Recognizing that customer attention is a limited resource, it then trains an autoregressive ranking policy with survival-weighted reward-to-go that prioritizes the most decision-relevant information early, so customers can find what they need with less effort. By decoupling utility learning from policy optimization, SMEO enables stable offline learning from biased logs and post-hoc media attribution without explicit media-level labels. Evaluated offline on large-scale e-commerce sessions using doubly robust off-policy estimation, SMEO improves estimated conversion by 5.5% and helps customers reach a purchase decision with 15% fewer swipes than existing baselines.

CCS CONCEPTS

• **Information systems** → **Learning to rank**; • **Computing methodologies** → **Reinforcement learning**.

KEYWORDS

E-Commerce, Multimodal Learning, Policy Optimization, Ranking

ACM Reference Format:
Prasenjit Dey, Frank McIntyre, and Arnab Sinha. 2026. Sequential Multimodal Evidence Optimization for Product Media Ranking in E-Commerce. In *Proceedings of the 35th ACM International Conference on Information and Knowledge Management (CIKM '26), November 07–11, 2026, Rome, Italy.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3799682.3840133

1 INTRODUCTION

In modern e-commerce, the product detail page is the primary storefront. Customers are presented with a media carousel: an ordered slate of heterogeneous assets such as product images, lifestyle photographs, demonstration videos, interactive 3D renders, and virtual try-on experiences. Due to mobile viewport constraints, these assets are not shown simultaneously; customers must actively swipe or click to reveal later slots. If early assets fail to capture attention or convey product value, customers may abandon the page before discovering decision-relevant media placed deeper in the carousel, as illustrated in Figure 1.

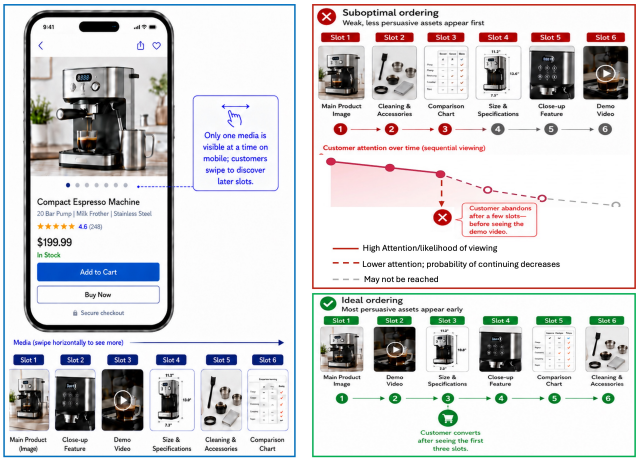

Figure 1: (Left) Product Detail page on Mobile. (Right) Suboptimal ordering may hide persuasive evidence until after customer drop-off, while ideal ordering front-loads decision-enabling media to support earlier conversion.

Because customers cannot physically inspect products online, purchase decisions depend on sequentially accumulating evidence from available media. This aligns with evidence accumulation models in cognitive science and decision theory [21]: customers consume informational signals until they either reach sufficient confidence to purchase or the interaction cost outweighs their interest. Therefore, learning the optimal ordering of multimodal product media to help customers find the information they need with less effort is a critical machine learning problem for e-commerce.

Prior studies show that high-quality product media improves conversion [15]. Despite its criticality, existing approaches in the literature rely on heuristic business rules or optimize for myopic engagement proxies (e.g., clicks, dwell time) that fail to guarantee terminal conversion [37]. For instance an eye-catching image may generate clicks, but fail to provide the functional specifications a customer needs to make an informed purchase decision. Furthermore, Learning-to-Rank (LTR) frameworks [5, 14] inherently assume items are independent, competing alternatives. In reality, product page media are *cooperative informational components*. The

CIKM '26, November 07–11, 2026, Rome, Italy

ACM ISBN 979-8-4007-2539-5/2026/11.
https://doi.org/10.1145/3799682.3840133

persuasive utility of a technical specification image is profoundly modulated by whether the customer first watched a lifestyle video establishing the product's value proposition.

Because cooperative media ranking introduces a fundamentally different learning problem, recent sequential slate optimizers like Seq2Slate [2] and listwise evaluators [18] prove inadequate. These methods expect dense, step-wise engagement over competing catalog items. Product media, by contrast, operates under severe reward sparsity where multiple assets jointly shape a single terminal purchase. Because media-level attribution is fundamentally unobserved in logs, standard methods naively assign equal credit to all engaged assets or rely on engagement proxies [13, 36]. This attribution crisis is compounded by *endogenous truncation*—users often abandon sessions early, causing naive sequence models to misattribute terminal outcomes to unobserved media deep in the slate. Finally, historical default policies create substantial position bias [35], confounding offline learning. These properties—cooperative assets, unobserved attribution, endogenous truncation, and strong position bias—make product media ranking a distinct problem. To address this, we present **Sequential Multimodal Evidence Optimization (SMEO)**, a two-stage offline framework that shifts the objective from ranking independent media items to optimizing the sequential accumulation of persuasive product evidence.

SMEO first learns a trajectory utility model from logged data using the consumed prefix—the media actually observed before customer drop-off—while using position-aware utility learning and survival-based depth balancing to reduce bias from historical media placement and variable stopping. It then freezes the utility model and trains an autoregressive pointer network [30] using survival-weighted reward-to-go optimization, where marginal utility gains are weighted by the probability that customers reach each position. This encourages the policy to surface the most decision-relevant media early, helping customers find what they need in fewer swipes.

The main contributions of this work are as follows:

(1) **Novel Problem Formulation:** We formulate product media ranking as sequential multimodal evidence optimization, where assets for the same product cooperatively influence purchase decisions under variable customer stopping depths.
(2) **Utility-Guided Sequencing:** We propose SMEO, a two-stage framework that learns conversion utility from consumed media prefixes under position-biased logs and trains an autoregressive policy to front-load decision relevant media.
(3) **Media Attribution:** The learnt utility model enables post-hoc contribution analysis via counterfactual masking, without requiring explicit media-level conversion labels.
(4) **Large Scale Offline Evaluation:** Evaluated with rigorous off-policy estimation, SMEO improves DR-CVR [9] by 5.5% and reduces swipes-to-conversion by 15% over the existing baseline.

2 RELATED WORK

Sequential Recommendation and Slate Optimization: Sequential recommenders such as DIN [37], DIEN [36], and SASRec [13] model user histories to predict future item interactions. Slate-ranking methods, including DLCM [1] and Seq2Slate [2], capture intra-list dependencies via listwise scoring [6] or autoregressive generation [8, 11]. However, these target catalog recommendation or click modeling, where items are competing alternatives and feedback is typically item-level engagement. In contrast, SMEO models cooperative media assets for a single product under sparse terminal label, optimizing sequentially viewed media rather than next-item engagement or full-slate relevance.

Credit Assignment and Counterfactual LTR: Attributing delayed outcomes to sequential interactions relates to Multi-Touch Attribution (MTA) [3, 10], which assigns conversion credit across marketing touchpoints but does not produce intra-page media-ordering policies. Counterfactual Learning to Rank (CLTR) [12, 16, 25] addresses exposure bias using propensity estimators, with extensions to sequential settings [31, 33]. SMEO is complementary: it does not claim full item-position identification, but reduces position-effect leakage via PAL[35] and handles endogenous carousel abandonment with survival-based depth balancing.

Multimodal Ranking and Offline Policy Learning: E-commerce systems increasingly use visual and textual embeddings for ranking [27, 34]. These systems typically treat media as independent product features rather than an ordered sequence of persuasive evidence. SMEO draws on offline policy learning [4, 7] and RLHF-style reward-model separation [17, 39], but adapts them to conversion-oriented media sequencing by decoupling trajectory utility learning from autoregressive ranking over biased offline logs.

3 PROBLEM FORMULATION

Let $\mathcal{A}$ denote the set of products. Each product $a \in \mathcal{A}$ is associated with a multimodal media catalog $\mathcal{M}^a = \{m_1^a, \ldots, m_{N_a}^a\}$, containing assets such as images, videos, and 3D renders. For a customer session s on product a_s, a ranking policy presents an ordered media slate $\pi^s \in \Pi(\mathcal{M}^{a_s})$, where $\Pi(\mathcal{M}^{a_s})$ denotes the set of feasible orderings of the product's media catalog. We write $\pi^s = [m^{a_s}_{\sigma^s(1)}, \ldots, m^{a_s}_{\sigma^s(N_{a_s})}]$, where $\sigma^s(t)$ denotes the index of the media asset placed at position t. Equivalently, $\pi_t^s = m^{a_s}_{\sigma^s(t)}$ denotes the asset at position t, and $\pi^s_{\leq t} = (\pi_1^s, \ldots, \pi_t^s)$ denotes the consumed prefix of length t. When the session is clear from context, we omit the superscript s.

Customers consume the slate sequentially and may stop before reaching the end. We therefore observe only a consumed prefix $\pi^s_{\leq T^s_{\text{obs}}}$, where T^s_{obs} is the endogenous stopping depth. Each logged session is represented as

$$s = \left(a_s, \pi^s_{\leq T^s_{\text{obs}}}, y^s\right),$$

where $y^s \in \{0, 1\}$ denotes the terminal purchase outcome. Media assets placed at positions $t > T^s_{\text{obs}}$ are not observed in the session and should not receive direct credit from the observed outcome.

Under this variable-depth, sparse-reward setting, we formulate product media ranking around two goals:

- **Optimal Media Sequencing:** Learn a product-conditioned ranking policy that selects an ordering $\pi^*(a) \in \Pi(\mathcal{M}^a)$ to maximize expected terminal conversion under customer stopping behavior:

$$\pi^*(a) = \underset{\pi \in \Pi(\mathcal{M}^a)}{\arg\max}\ \mathbb{E}_{T \sim P_{\text{stop}}(\cdot \mid a, \pi)}\left[P(y = 1 \mid a, \pi_{\leq T})\right]. \quad (1)$$

- **Latent Media Contribution Estimation:** Estimate the marginal contribution of each media asset to terminal conversion, while accounting for the fact that media-level conversion labels

are unobserved and contribution may depend on the surrounding media context and ordering.

This formulation differs from standard listwise ranking: the reward is sparse and terminal, the consumed trajectory is endogenously truncated, and media-level contribution is latent, context-dependent, and unobserved in logged data.

4 PROPOSED METHODOLOGY

Figure 2 summarises the SMEO framework. We first construct multimodal asset representations (Section 4.1), then learn a trajectory utility model U from logged consumed prefixes under sparse terminal supervision (Section 4.2). We next train an autoregressive pointer network policy that uses the frozen U as a utility-guided reward model to generate conversion-maximising media orderings (Sections 4.3–4.4). Finally, we use the frozen utility model for offline media contribution analysis (Section 4.5).

4.1 Multimodal Asset Representation

Each media asset $m_i^a \in \mathcal{M}^a$ is encoded into a unified token $\mathbf{x}_i \in \mathbb{R}^d$ that fuses visual content, modality type, and interaction signals.

4.1.1 Visual Content Embedding. Let $\kappa_i \in \mathcal{K} = \{\texttt{img}, \texttt{video}, \texttt{3d}\}$ denote the modality type of asset m_i^a. Visual content is encoded modality-conditionally:

$$\mathbf{v}_i = \begin{cases} f_{\text{ViT}}(m_i^a) & \kappa_i = \texttt{img} \\ f_{\text{VMAE}}(m_i^a) & \kappa_i = \texttt{video} \\ f_{\text{Uni3D}}(m_i^a) & \kappa_i = \texttt{3d} \end{cases} \in \mathbb{R}^{d_v}, \tag{2}$$

where f_{ViT} is a frozen ViT-L/14 from CLIP [20], f_{VMAE} is a frozen VideoMAE encoder [28] that mean-pools frame-level features, and f_{Uni3D} is a frozen Uni3D foundation model [38] that encodes 3D point clouds and VTO meshes directly into the shared vision-language latent space, with all outputs projected to $d_v = 256$. A learnable modality-type embedding $\mathbf{e}_{\kappa_i} \in \mathbb{R}^{d_e}$ ($d_e = 32$) captures distinct persuasive roles of each media type grounded in media function.

4.1.2 Interaction Features and Train-Serve Decoupling. Beyond visual semantics, we incorporate two numerical signal groups. First, a **historical prior vector** $\mathbf{h}_i \in \mathbb{R}^{d_h}$ captures aggregate asset-level statistics from past logs, such as average dwell time and click-through rates etc.; these priors are available at both training and inference time. Second, during training only, we extract in-session interaction signals $\boldsymbol{\tau}_i = [\log(1+d_i), \tilde{k}_i, z_i] \in \mathbb{R}^{d_\tau}$, where d_i is normalized dwell time, $\tilde{k}_i$ is click count, and z_i is zoom count. These signals provide direct gradient signal linking visual representations to observed engagement quality, but are unavailable at page-load inference.

To bridge this train–inference gap, we apply feature masking [23]: during training, $\boldsymbol{\tau}_i$ is replaced by a learned mask token $\mathbf{m}_\tau \in \mathbb{R}^{d_\tau}$ with probability $p_{\text{drop}} = 0.3$, and always masked at inference. This exposes the shared encoder to two regimes: interaction-visible examples provide engagement supervision, while masked examples force prediction from visual content and historical priors alone. The encoder thus learns engagement-correlated representations that remain calibrated without in-session interactions; the learned mask token also preserves activation statistics under layer normalization.

4.1.3 Composite Token Representation. The final token for asset i at consumption position t fuses visual, modality, historical, and in-session embeddings:

$$\mathbf{x}_i^{(t)} = \text{LN}\Big(W_v\mathbf{v}_i + W_e\mathbf{e}_{\kappa_i} + W_h\mathbf{h}_i + W_\tau\tilde{\boldsymbol{\tau}}_i + \mathbf{p}_t\Big) \in \mathbb{R}^d, \tag{3}$$

where $\tilde{\boldsymbol{\tau}}_i \in \{\boldsymbol{\tau}_i, \mathbf{m}_\tau\}$ is the conditionally masked interaction vector, $W_{v,e,h,\tau}$ are learnable linear projections, $\mathbf{p}_t$ is a sinusoidal positional embedding [29] capturing evidence-accumulation order within the slate, LN denotes layer normalisation, and $d = 256$. These position-aware tokens form the observed prefix sequence consumed by the trajectory utility model.

4.2 Trajectory Utility Model

4.2.1 Causal Sequence Encoder and Position-Aware Learning. We parameterise the trajectory utility function as an L-layer Transformer encoder [29] with *causal* left-to-right self-attention masking. For a consumed prefix $\pi^s_{\le t}$, the encoder produces contextualised hidden states representing cumulative evidential belief:

$$\big[\mathbf{h}_1, \ldots, \mathbf{h}_t\big] = \text{TransEnc}_{\text{causal}}\Big(\big[\mathbf{x}^{(1)}_{\sigma(1)}, \ldots, \mathbf{x}^{(t)}_{\sigma(t)}\big]\Big). \tag{4}$$

To reduce leakage of systematic position effects into the content encoder, we employ a Position-Aware Learning (PAL) architecture [35]. We decompose the utility logit into a content-driven component from the Transformer state and a position-only component from a shallow network that observes only positional embeddings:

$$s_{\text{content}} = \text{MLP}_{\text{content}}(\mathbf{h}_t), s_{\text{pos}} = \text{MLP}_{\text{pos}}([\mathbf{p}_1, \ldots, \mathbf{p}_t]). \tag{5}$$

During utility-model training, the prediction uses the sum of both logits: $U_{\text{train}}(\pi^s_{\le t}) = \text{sigmoid}(s_{\text{content}} + s_{\text{pos}})$.

The position-only tower acts as a nuisance pathway for systematic slot-depth effects, reducing position-bias leakage into the sequence encoder. During training, it receives only consumed-prefix positional embeddings, with 10% dropout to limit over-reliance on slot features. During policy optimization and inference, the PAL tower is discarded, and the frozen reward model uses the PAL-adjusted content utility, as shown in Fig. 2:

$$U(\pi^s_{\le t}) = \text{sigmoid}(s_{\text{content}}) \in [0, 1]. \tag{6}$$

This mitigates position-bias leakage from biased logged data.

4.2.2 Variable-Depth Supervision. The utility model is trained on logged sessions $\mathcal{D} = \{(a_s, \pi^s_{\le T^s_{\text{obs}}}, y^s)\}_{s=1}^{|\mathcal{D}|}$. The observed media evidence available to influence y^s is restricted to the consumed prefix: assets beyond T^s_{obs} were not viewed and should not receive direct credit for the purchase outcome. Each session therefore supervises the model only at its observed consumed depth:

$$\mathcal{L}_U = -\frac{1}{|\mathcal{D}|}\sum_{s\in\mathcal{D}} \hat{w}^s\Big[y^s \log U_{\text{train}}\big(\pi^s_{\le T^s_{\text{obs}}}\big) + (1-y^s)\log\big(1-U_{\text{train}}\big(\pi^s_{\le T^s_{\text{obs}}}\big)\big)\Big], \tag{7}$$

where $\hat{w}^s$ is a trajectory balancing weight (Section 4.2.3).

Remark (why all-prefix supervision is incorrect). A natural alternative augments $\mathcal{L}_U$ with $\text{BCE}(U_{\text{train}}(\pi^s_{\le t}), y^s)$ for every $t < T^s_{\text{obs}}$. For example, in a session

$$[\texttt{weak\_img}, \texttt{weak\_img}, \texttt{great\_video}] \to y = 1,$$

supervising $U_{\text{train}}(\pi_{\le 1})$ toward 1 can incorrectly credit the weak image for a conversion observed only after the later video was

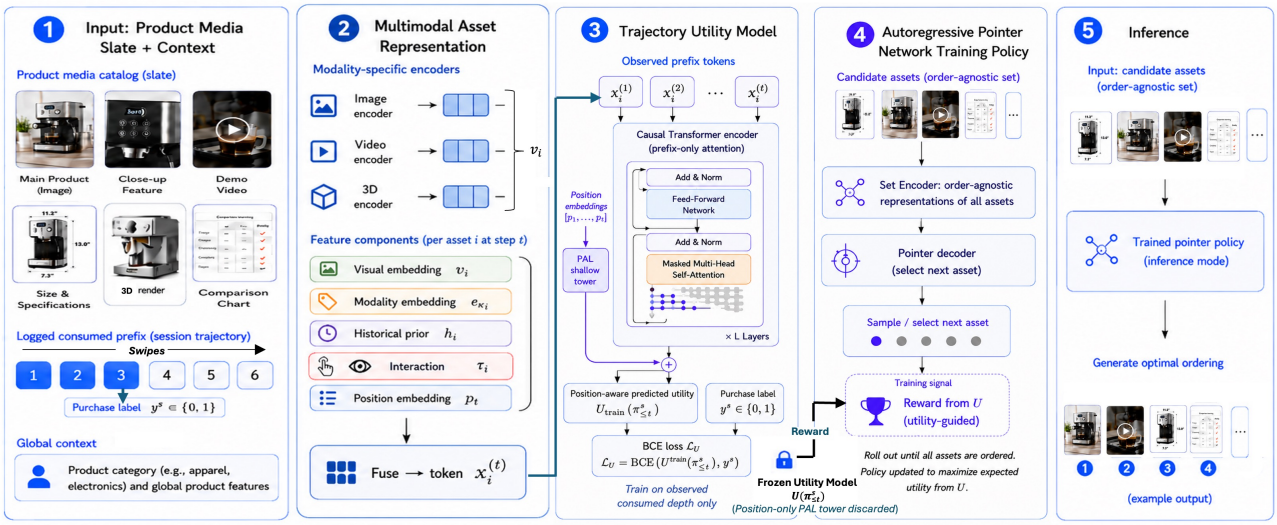

Figure 2: Complete end-to-end SMEO Framework

consumed. Equation 7 avoids this leakage by supervising each session only at its observed consumed depth.

4.2.3 Stopping Depth Balancing. Product detail page logs are dominated by shallow prefixes and relatively few deep prefixes because customers often stop before reaching later media, causing naive training to underfit deeper consumed trajectories. Unlike standard LTR, which corrects item-level examination bias for pointwise rewards [12], our utility model predicts terminal conversion for an observed media trajectory. We therefore use survival-based depth reweighting to balance observations across stopping depths:

$$\hat{\rho}_t = \frac{\#\{s \in \mathcal{D} : T^s_{\text{obs}} \geq t\}}{|\mathcal{D}|}, \qquad \hat{w}^s = \min\left(\frac{1}{\hat{\rho}_{T^s_{\text{obs}}}}, W_{\max}\right), \quad W_{\max} = 10.$$

Here, $\hat{\rho}_t$ is the empirical probability of reaching depth t, and $\hat{w}^s$ upweights rare deep prefixes while clipping controls variance.

4.3 Autoregressive Pointer Network Policy

After training, U from Eq. 6 is frozen and used only to score candidate prefixes during policy optimization. We parameterise the ranking policy p_θ as a pointer network [30] — an autoregressive model whose output vocabulary is the input set itself. Three properties motivate this choice over a standard decoder. First, it generalises across products without retraining, operating on asset *embeddings* rather than fixed item IDs. Second, autoregressive masked softmax guarantees each asset appears exactly once in the output. Third, it scales naturally to variable slate sizes $N_a \in [1, N_{\max}]$.

4.3.1 Set Encoder. The encoder produces order-agnostic representations of all assets:

$$\left[\bar{\mathbf{h}}_1, \ldots, \bar{\mathbf{h}}_{N_a}\right] = \text{TransEnc}_{\text{full}}([\mathbf{x}^{(0)}_1, \ldots, \mathbf{x}^{(0)}_{N_a}])$$

where $\mathbf{x}^{(0)}_i$ is constructed using the learned interaction mask token $\mathbf{m}_\tau$ and no positional embedding. Removing positional priors ensures representations depend strictly on content and mutual context, treating the catalog as an unordered set.

4.3.2 Pointer Decoder. The decoder constructs the ordering $\pi = [m_{\sigma(1)}, \ldots, m_{\sigma(N_a)}]$ autoregressively using a 2-layer causal Transformer decoder. At step t, the decoder attends over the sequence of previously selected encoder representations $[\bar{\mathbf{h}}_{\sigma(1)}, \ldots, \bar{\mathbf{h}}_{\sigma(t-1)}]$ via causal self-attention, producing a decoder state $\mathbf{s}_t \in \mathbb{R}^d$. Each unselected asset m_i is then scored via pointer attention:

$$z^{(t)}_i = \mathbf{v}^\top \tanh(W_1 \bar{\mathbf{h}}_i + W_2 \mathbf{s}_t), \quad i \notin \text{sel}_{<t},$$

and selected according to

$$p_\theta(m_{\sigma(t)} \mid \pi_{<t}) = \text{softmax}\left(\{z^{(t)}_i\}_{i \notin \text{sel}_{<t}}\right), \text{ where } \mathbf{v}, W_1, W_2$$

are learnable projections and $\mathbf{s}_1$ is a learnable intital state. Masked softmax over unselected assets guarantees a valid permutation.

4.4 Policy Optimization with Survival-Weighted Reward-to-Go

Here, U denotes the frozen content-based utility from Eq. (6), with the PAL tower removed. Generated orderings are scored in the same pre-serving setting as inference: in-session interactions are replaced by the learned mask token, so policy optimization uses only information available before carousel interaction. For a generated ordering $\pi = [m_{\sigma(1)}, \ldots, m_{\sigma(N_a)}]$, we define the incremental utility of the asset placed at position t as

$$\Delta_t(\pi) = U(\pi_{\leq t}) - U(\pi_{\leq t-1}),$$

where $U(\pi_{\leq 0})$ is initialized to the empirical base conversion rate. This marginal formulation removes the constant base utility: an uninformative asset yields $\Delta_t \approx 0$, preventing the policy from accumulating reward merely from longer prefixes.

A natural objective is the expected utility of the prefix actually reached by the customer, $\mathbb{E}_T[U(\pi_{\leq T})]$. Using cumulative marginal gains, this decomposes as

$$\mathbb{E}_T[U(\pi_{\leq T})] = U(\pi_{\leq 0}) + \sum_{t=1}^{N_a} \Pr(T \geq t) \Delta_t(\pi).$$

Thus, the marginal utility at position t should be weighted by the survival probability of that position being reached, rather than by the probability of stopping exactly at t. Estimating $\hat{\rho}_t = \Pr(T_{\text{obs}} \geq t)$ from logged session depths, we define the step reward as

$$r_t(\pi) = \hat{\rho}_t \gamma^{t-1} \Delta_t(\pi). \tag{8}$$

where discount factor $\gamma \in (0, 1]$ optionally emphasizes earlier persuasive evidence. Since $\hat{\rho}_t$ decreases with depth on mobile carousels, utility gains obtained at early positions receive larger reward than identical gains placed later. The policy is therefore explicitly optimized to surface conversion-informative media before likely customer drop-off, which directly targets reduced swipe depth. We optimize the autoregressive policy p_θ using reward-to-go (RTG) REINFORCE [24, 32]. For a sampled ordering π, the return from position t onward is:

$$G_t(\pi) = \sum_{j=t}^{N_a} r_j(\pi).$$

Unlike terminal REINFORCE, which assigns the same slate reward to every action, RTG provides a temporally localized credit signal by assigning each decoding decision only its future rewards. To reduce variance, we use a self-critical sequence training (SCST) baseline [22]. For each product slate, we compute a greedy ordering π^{gr} using argmax decoding from the current policy and define

$$A_t(\pi) = G_t(\pi) - G_t(\pi^{\text{gr}}).$$

The greedy baseline is treated as stop-gradient and computed by deterministic decoding under the current policy. This yields a low-cost policy-dependent baseline requiring one additional rollout per batch, which is practical in our setting where N_a is small.

The entropy-regularized policy loss is

$$\mathcal{L}_\pi = -\sum_{t=2}^{N_a} \log p_\theta(m_{\sigma(t)} \mid \pi_{<t}) \, \text{sg}\Big(A_t^{(k)}\Big) - \lambda_H \sum_{t=2}^{N_a} \mathcal{H}(p_\theta(\cdot \mid \pi_{<t})),$$

where $\text{sg}(\cdot)$ denotes the stop-gradient operator and $\mathcal{H}$ is the Shannon entropy. **Note:** The summation starts at $t = 2$ because the primary product image is conventionally fixed at $t = 1$, a common product-page UX constraint. SMEO therefore optimizes only the subsequent slots conditional on this fixed initial state, ensuring generated orderings respect layout conventions. The entropy bonus ($\lambda_H = 5 \times 10^{-4}$) prevents premature policy collapse while offline advantages are noisy. The pointer network restricts actions to valid product media and masks already selected assets, preventing invalid or duplicate outputs. We tune λ_H and monitor held-out metrics to detect potential reward over-optimization.

4.5 Media Contribution Estimation

The frozen content-based utility model also enables offline media diagnostics. Here, U denotes the utility model with the PAL tower removed and in-session interactions masked. Given ordering π, we estimate media asset (m_j^a) contribution via leave-one-out masking:

$$\varphi_{\text{LOO}}(m_j^a \mid \pi) = U(\pi) - U(\pi \setminus \{m_j^a\}), \tag{9}$$

where $\pi \setminus \{m_j^a\}$ removes m_j^a and re-indexes the remaining assets. This context- and ordering-dependent score helps identify high value or redundant media and whether specific media types should be surfaced earlier. We also compute evidence accumulation curves, $E_t(\pi) = U(\pi_{\leq t})$, to visualize how quickly an ordering builds conversion utility before likely drop-off. These diagnostics, presented in Section 5.4, provide category audit and merchandising teams in e-commerce with interpretable signals for prioritizing, replacing, or requesting media assets. These are interpretability diagnostics; primary evaluation uses observed conversion-oriented metrics.

5 EXPERIMENTS

We investigate the following research questions:

- **RQ1:** Does SMEO improve conversion and swipe efficiency?
- **RQ2:** Which SMEO components drive the gains?
- **RQ3:** Does SMEO learn interpretable media-ordering behavior?

5.1 Experimental Setup

Dataset: We utilize a large-scale, proprietary dataset comprising of anonymized mobile session logs from a global e-commerce store spanning several weeks across multiple product categories (electronics, apparel, home, beauty): approximately 150M randomly sampled sessions over millions of unique products with tens of millions of total media assets (static images, videos, and 3D interactive renders). Each session records the product identifier, ordered media slate, per-asset interaction signals (dwell time, click and zoom counts), observed stopping depth T_{obs}^s, and terminal binary purchase outcome y^s. We evaluate on a temporally held-out test set to prevent data leakage. Candidate slates are capped at $N_{\max} = 15$.

Experimental Details: All models are trained on 8× NVIDIA V100 GPUs with frozen, precomputed visual encoders. The trainable architecture—multimodal projections, trajectory utility model, PAL tower, and pointer policy—totals ∼4.5M parameters. The trajectory encoder uses L=4 causal Transformer layers (H=4 heads, d=256); the pointer policy uses a 2-layer autoregressive decoder with identical dimensionality. We optimize with AdamW, lr= 10^{-4}, 10% warmup, and cosine decay. The utility model trains for 15 epochs with early stopping on validation BCE. The pointer policy subsequently trains for 10 epochs using the frozen U as a dense reward signal. To stabilize policy optimization, we apply per-batch advantage normalization, survival discount γ=0.90, entropy regularisation $\lambda_H = 5{\times}10^{-4}$ and warm-start the decoder via supervised pre-training on logged historical orderings for 2 epochs.

Baselines: We compare SMEO against five baselines:

(1) **Heuristic:** a rule-based ordering that serves as our real-world anchor baseline.
(2) **CTR:** a pointwise model that optimizes on per asset engagement proxy (clicks) rather than terminal conversion.
(3) **CLTR-CVR:** a pointwise conversion model trained with standard Counterfactual LTR [12]. It assigns the terminal session outcome y^s to each viewed asset independently and corrects for position bias using Inverse Propensity weighting. This tests whether item-level debiasing suffices without sequential inter-asset context, while retaining the conversion objective.
(4) **Listwise-MM:** a multimodal listwise ranker [1, 18] that predicts terminal conversion from a self-attended slate representation. This retains inter-asset context but does not model consumed-prefix utility or autoregressive decoding.
(5) **Seq2Slate-Adapt:** an autoregressive pointer network [2] with the same policy architecture as SMEO, trained using terminal

slate-level REINFORCE on the session outcome y^s with an SCST baseline [22]. This isolates the value of SMEO's consumed-prefix utility learning and survival-weighted RTG objective.

All learned baselines use the same features as SMEO, so differences reflect modeling choices rather than feature access.

Metrics. We evaluate offline conversion value and early-swipe efficiency. The terminal label $y^s \in \{0, 1\}$ denotes whether session s ended in purchase. Although logged ordering is not randomized, it varies across categories due to overlapping business rules, catalog updates, and upstream experiments. In our data, a large majority of products have multiple historical media permutations, providing partial support for off-policy evaluation. We therefore restrict estimates to supported regions and use clipping to control variance.

SNIPS-CVR. We report clipped self-normalized IPS-CVR [26]:

$$\widehat{\mathrm{CVR}}_{\mathrm{SNIPS}}(\pi) = \frac{\sum_s \bar{w}_s y^s}{\sum_s \bar{w}_s}, \qquad \bar{w}_s = \min\left(\frac{\hat{p}_\pi(\pi^s \mid a_s)}{\hat{p}_\mu(\pi^s \mid a_s)}, W_{\mathrm{IPS}}\right),$$

Here, π^s denotes the logged ordering in session s, $\pi(a_s)$ denotes the ordering induced by the evaluated target policy π for product a_s. $\hat{p}_\mu$ is a behavior-cloned logging model, and $\hat{p}_\pi$ is the stochastic distribution induced by the target policy. For deterministic rankers, $\hat{p}_\pi$ is computed using a Plackett–Luce distribution over model scores [19]; pointwise models sort by score, Listwise-MM greedily decodes marginal gain, and pointer models decode autoregressively.

DR-CVR. We also report doubly robust CVR [9, 25]:

$$\widehat{\mathrm{CVR}}_{\mathrm{DR}}(\pi) = \frac{1}{|\mathcal{D}|} \sum_s \left[\hat{r}(a_s, \pi(a_s)) + \bar{w}_s \left(y^s - \hat{r}(a_s, \pi^s)\right)\right],$$

where $\hat{r}(a, \pi)$ is a K-fold cross-fitted predictor of $\mathbb{E}[y \mid a, \pi]$ from product features and media ordering. DR-CVR combines direct reward prediction with logged-policy correction.

MSC. For converted sessions, let $r_i^s = \mathbb{I}\{m_i \text{ is engaged in } s,\ y^s = 1\}$. For ordering $\pi(a_s)$, where $\pi_k(a_s)$ is the asset placed at rank k:

$$\mathrm{MSC} = \mathrm{median}_{s: y^s=1}\left[\min\{k : r^s_{\pi_k(a_s)} = 1\}\right].$$

MSC measures how early a policy surfaces historically conversion-associated media; lower is better. Since stopping behavior may change under a new ordering, MSC is not a causal swipe estimate but a relative ordering-quality metric for early evidence placement.

5.2 RQ1: SMEO vs. Baselines

Table 1 shows three trends. **First, engagement proxies are insufficient:** optimizing CTR yields only marginal conversion lift and almost no swipe-efficiency improvement, indicating that attention-grabbing media does not necessarily provide purchase-decisive evidence. **Second, conversion objective and context both matter:** CLTR-CVR improves DR-CVR over engagement ranking, but remains limited by independent asset scoring and unmodeled sequence redundancy. Listwise-MM adds inter-asset context, producing an additional +1.6 point relative DR-CVR gain over CLTR-CVR and a larger MSC reduction. **Finally, trajectory-aware reward design is most effective:** SMEO improves over Seq2Slate-Adapt by +1.9 points in relative DR-CVR and 4.5 points in MSC reduction. While both use autoregressive decoding, SMEO improves beyond Seq2Slate-Adapt by restructuring the reward signal: survival-weighted RTG credits marginal utility gained before likely drop-off, directly optimizing early evidence delivery.

Table 1: For confidentiality, all metrics are reported as relative change against the heuristic baseline. Values include 95% confidence intervals. Higher is better for SNIPS-CVR, DR-CVR; lower is better for MSC.

Method	ΔSNIPS-CVR ↑	ΔDR-CVR ↑	ΔMSC ↓
Heuristic (Rule-based)	–	–	–
Engagement(CTR)	+0.7±0.5%	+0.4±0.3%	−0.8±0.5%
CLTR-CVR	+1.5±0.7%	+1.1±0.5%	−3.6±0.8%
Listwise-MM	+3.2±0.6%	+2.7±0.4%	−8.5±1.2%
Seq2Slate-Adapt	+4.1±0.5%	+3.5±0.4%	−10.8±1.3%
SMEO	**+6.1±0.6%**	**+5.4±0.3%**	**−15.3±1.5%**

Table 2: Ablation study of SMEO components. Values are relative to full SMEO.

Variant	ΔSNIPS-CVR ↑	ΔDR-CVR ↑	ΔMSC ↓
SMEO full	–	–	–
w/o PAL	−1.2±0.4%	−1.0±0.3%	+2.1±0.7%
PAL tower kept in U	−1.8±0.5%	−1.5±0.4%	+3.4±0.8%
w/o survival weighting	−2.3±0.5%	−1.9±0.3%	+4.8±0.9%
w/o interaction masking	−0.8±0.4%	−0.7±0.3%	+1.5±0.6%

5.3 RQ2: Ablation Study

We ablate SMEO-specific components in Table 2 and report relative changes against full SMEO using the same offline evaluation protocol.

Effect of Position-Aware Learning: Removing the PAL tower during Stage 1 training (*w/o PAL*) forces the utility model to absorb historical position effects into the content encoder, reducing downstream DR-CVR by 1.0% relative to full SMEO.

Importance of Removing PAL at Policy Time: Training with PAL but keeping the PAL tower during Stage 2 policy optimization (*PAL tower kept in U*) further worsens swipe efficiency (ΔMSC = +3.4%). This suggests that exposing the policy to position-biased reward components encourages it to exploit learned slot effects rather than optimize content-driven sequencing.

Survival Weighting Drives Early Placement: Removing survival weighting causes the largest MSC degradation, supporting customer reach probability as the key mechanism for front-loading persuasive evidence. This also helps explain SMEO's advantage over Seq2Slate-Adapt: autoregressive decoding alone is insufficient without reach-weighted marginal utility.

Train–Serve Robustness: Removing in-session interaction masking (*w/o interaction masking*) slightly degrades CVR and swipe efficiency. This indicates that masking improves robustness under page-load inference, where real-time interaction features are unavailable.

5.4 RQ3: Qualitative Media Diagnostics

Evidence Front-Loading: Figure 4 plots prefix-level predicted conversion utility as media are revealed slot by slot. In the left panel, this is $E_t(\pi) = U(\pi_{\le t})$, the trajectory utility score for the first t assets; SMEO rises fastest, as expected since it optimizes U. To avoid circular evaluation, the right panel scores the same prefixes

(a) Early Evidence Contribution: (top) Heuristic baseline ranking vs (bottom) SMEO. **(b) Contextual Redundancy**

Figure 3: (a) With the main product image fixed at slot 1, SMEO moves high-information assets earlier. Demo videos for coffee makers and nutrition labels for supplements produce larger LOO contribution when surfaced near the front. (b) For sneakers, near-identical views drive $\Delta_t = U(\pi_{\leq t}) - U(\pi_{\leq t-1})$ toward zero after the first image, while diverse assets sustain marginal utility by adding complementary evidence.

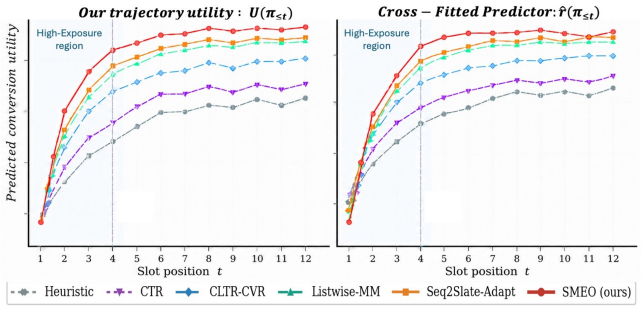

Figure 4: Evidence accumulation curves across prefix depth.

with the independent cross-fitted predictor $\hat{r}(a, \pi_{\leq t})$ from Sec. 5. SMEO still leads through the early slots, before likely customer drop-off, while Listwise-MM and Seq2Slate-Adapt partially catch up later. This indicates that SMEO primarily improves *earlier* placement of conversion-informative media rather than merely increasing full-slate utility. Fig. 3(a) provides qualitative evidence: with slot 1 fixed by the standard product-page convention, moving the same high-information asset earlier yields larger LOO (Eqn. 9) contribution and higher $U(\pi_{\leq 3})$, showing that ordering affects both marginal media value and early evidence accumulation.

Redundancy correlation: We quantify redundancy by correlating each asset's marginal gain, $\Delta_t = U(\pi_{\leq t}) - U(\pi_{\leq t-1})$, with its maximum visual similarity to the preceding prefix, $s_t^{\max} = \max_{k<t} \cos(\mathbf{v}_{\sigma(t)}, \mathbf{v}_{\sigma(k)})$, where $\mathbf{v}_{\sigma(t)}$ is the visual embedding of the asset at position t. To remove position-depth confounding, we compute partial Spearman correlation controlling for t. Strong negative correlation ($\boldsymbol{\rho_{\text{partial}} = -0.61}$, $\boldsymbol{p < 0.001}$) shows that visually redundant assets contribute less incremental utility (Fig. 3(b)).

Category-Level Attribution Patterns: Aggregating φ_{LOO} by modality and category shows that SMEO learns category-specific media roles that are meaningful for e-commerce merchandising. **3D assets contribute significantly more in spatial categories such as furniture and home fixtures** than elsewhere (0.091±0.019 vs. 0.028±0.011, p<0.01), where customers need geometry, scale, and room-fit evidence. **Videos dominate dynamic categories such as appliances and activewear** (0.103±0.024 vs. 0.041±0.016, p<0.001), where operation, motion, or fit-in-use must be demonstrated. **Specification and ingredient images contribute most in high-scrutiny categories such as electronics and consumables** (0.078±0.017 vs. 0.022±0.009, p<0.01), where factual details reduce purchase uncertainty. This confirms that SMEO learns media-type-specific persuasion roles from conversion signal alone,without any supervision. This enables offline decisions about which assets to prioritize, audit, or request for each product category, and provides actionable guidance for seller content improvement in e-commerce.

6 BATCH INFERENCE AND SERVING

SMEO is designed as an offline ranking system decoupled from real-time page serving. In a representative deployment configuration, media orderings are generated from static product-level signals — visual embeddings, historical priors, and product/media metadata— with no dependence on real-time customer state at page load. SMEO uses a distributed batch pipeline with precomputed embeddings and cached features. The short decode length ($N_{\max}$=15) makes generation tractable: in our experiments, a full refresh on a 16-GPU V100 cluster completes in a few hours, with inference accounting for less than half of wall-clock time. The pipeline scales horizontally — refresh cost is linear in catalogue size — and supports catalogue-scale operation through periodic incremental refreshes. Generated orderings are written to a distributed key-value store keyed by product identifier and retrieved at request time through a single point lookup (sub-5 ms latency). When an ordering is unavailable, the system falls back to a default ordering, preserving availability.

7 CONCLUSION

We introduced SMEO, a two-stage framework that treats product media as sequential, cooperative evidence for purchase decisions. By learning consumed-prefix conversion utility and optimizing a survival-weighted autoregressive policy, SMEO surfaces the most decision-relevant media early, before likely customer drop-off. Offline evaluation shows consistent gains over heuristic, pointwise, listwise, and autoregressive baselines in estimated conversion and swipe efficiency. Diagnostics further reveal redundancy-aware gains and category-specific media contributions. Online exploration and personalization are natural extensions.

8 GEN-AI USAGE DISCLOSURE

In preparing this manuscript, generative AI tools were used solely for language refinement—including improving grammar, style, clarity, and readability of text that was originally authored by the paper's authors. No sections of the manuscript, including the abstract, related work, methodology, experimental results, or conclusions, were generated by AI systems. All conceptual contributions, technical content, experimental design, data analysis, and interpretations are the authors' own original work. The authors accept full responsibility for the correctness and integrity of the manuscript.

REFERENCES

[1] Qingyao Ai, Keping Bi, Jiafeng Guo, and W. Bruce Croft. 2018. Learning a Deep Listwise Context Model for Ranking Refinement. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '18)*. ACM, 135–144. https://doi.org/10.1145/3209978.3209985

[2] Irwan Bello, Sayali Kulkarni, Sagar Jain, Craig Boutilier, Ed Chi, Elad Eban, Xiyang Luo, Alan Mackey, and Ofer Meshi. 2019. Seq2Slate: Re-ranking and Slate Optimization with RNNs. arXiv:1810.02019 [cs.IR] https://arxiv.org/abs/1810.02019

[3] Ron Berman. 2018. Beyond the Last Touch: Attribution in Online Advertising. *Marketing Science* 37, 5 (2018), 771–792. https://doi.org/10.1287/mksc.2018.1104

[4] David Brandfonbrener, Alberto Bietti, Jacob Buckman, Romain Laroche, and Joan Bruna. 2023. Offline RL with Noe and Inverse Propensity Scoring. In *International Conference on Learning Representations (ICLR 2023)*. https://openreview.net

[5] Christopher JC Burges. 2010. From ranknet to lambdarank to lambdamart: An overview. *Learning* 11, 23-581 (2010), 81.

[6] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to Rank: From Pairwise Approach to Listwise Approach. In *Proceedings of the 24th International Conference on Machine Learning (ICML '07)*. ACM, 129–136. https://doi.org/10.1145/1273496.1273513

[7] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. 2021. Decision Transformer: Reinforcement Learning via Sequence Modeling. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, Vol. 34. Curran Associates, Inc., 15084–15097. https://neurips.cc

[8] Romain Deffayet, Thibaut Thonet, Jean-Michel Renders, and Maarten de Rijke. 2023. Generative Slate Recommendation with Reinforcement Learning. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (WSDM '23)*. ACM, 781–789. https://doi.org/10.1145/3539597.3570412

[9] Miroslav Dudík, John Langford, and Lihong Li. 2011. Doubly Robust Policy Evaluation and Learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML '11)*. Omnipress, 1097–1104. https://arxiv.org/abs/1103.4601

[10] Kan Ren et al. 2018. Learning Multi-touch Conversion Attribution with Dual-attention Mechanisms for Online Advertising. In *CIKM '18*. https://doi.org/10.1145/3269206.3271781

[11] Ray Jiang, Sven Gowal, Timothy A. Mann, and Danilo Jimenez Rezende. 2018. Beyond Greedy Ranking: Slate Optimization via List-CVAE. In *Proceedings of the 35th International Conference on Machine Learning (ICML '18) (Proceedings of Machine Learning Research, Vol. 80)*. PMLR, 2330–2339. https://mlr.press

[12] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased learning-to-rank with biased feedback. In *Proceedings of the tenth ACM international conference on web search and data mining*. 781–789.

[13] Wang-Cheng Kang and Julian McAuley. 2018. Self-Attentive Sequential Recommendation. In *Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM)*. IEEE Computer Society, 197–206. https://doi.org/10.1109/ICDM.2018.00035

[14] Tie-Yan Liu. 2009. Learning to Rank for Information Retrieval. *Foundations and Trends® in Information Retrieval* 3, 3 (2009), 225–331. https://doi.org/10.1561/1500000016

[15] Nfinite. 2023. *The Only Stats You Need for Visuals in eCommerce 2024*. Industry Report. Nfinite. https://nfinite.app

[16] Harrie Oosterhuis. 2023. Doubly-Robust Estimation for Correcting Position-Bias in Click Feedback for Unbiased Learning-to-Rank. *ACM Transactions on Information Systems (TOIS)* 41, 3 (2023), 1–33. https://doi.org/10.1145/3569453

[17] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems* 35 (2022), 27730–27744.

[18] Changhua Pei, Yi Zhang, Yongfeng Zhang, Fei Sun, Xiao Lin, Hanxiao Sun, Jian Wu, Peng Jiang, Junfeng Ge, Wenwu Ou, et al. 2019. Personalized re-ranking for recommendation. In *Proceedings of the 13th ACM conference on recommender systems*. 3–11.

[19] Robert L. Plackett. 1975. The Analysis of Permutations. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 24, 2 (1975), 193–202. https://doi.org/10.2307/2346567

[20] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, and Ilya Sutskever et. al. 2021. Learning Transferable Visual Models from Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML '21)*. PMLR. https://mlr.press

[21] Roger Ratcliff and Gail McKoon. 2008. The diffusion decision model: theory and data for two-choice decision tasks. *Neural computation* 20, 4 (2008), 873–922.

[22] Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-Critical Sequence Training for Image Captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '17)*. IEEE Computer Society, 7008–7024. https://doi.org/10.1109/CVPR.2017.125

[23] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* 15, 56 (2014), 1929–1958. http://jmlr.org

[24] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction* (second ed.). The MIT Press, Cambridge, Massachusetts. http://incompleteideas.net/book/the-book-2nd.html

[25] Adith Swaminathan and Thorsten Joachims. 2015. Counterfactual Risk Minimization: Learning from Logged Bandit Feedback. In *Proceedings of the 32nd International Conference on Machine Learning (ICML '15) (JMLR Workshop and Conference Proceedings, Vol. 37)*. JMLR.org, 555–563. https://mlr.press

[26] Adith Swaminathan and Thorsten Joachims. 2015. The self-normalized estimator for counterfactual learning. *advances in neural information processing systems* 28 (2015).

[27] Yuan Tang, Sheng Chen, Sihui Wang, and Jianpeng Xu. 2024. Multimodal Learning for Click-Through Rate Prediction: A Survey. *arXiv preprint arXiv:2402.04321* (2024). arXiv:2402.04321 [cs.IR] https://arxiv.org

[28] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. 2022. VideoMAE: Masked Autoencoders are Data-Efficient Learners for Self-Supervised Video Pre-Training. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, Vol. 35. Curran Associates, Inc., 10078–10093. https://neurips.cc

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Ilya Polosukhin. 2017. Attention is All You Need. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*. Curran Associates, Inc., 5998–6008. https://neurips.cc

[30] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. *Advances in neural information processing systems* 28 (2015).

[31] Zhenlei Wang, Shiqi Shen, Zhipeng Wang, Bo Chen, Xu Chen, and Ji-Rong Wen. 2022. Unbiased Sequential Recommendation with Latent Confounders. In *Proceedings of the ACM Web Conference 2022 (WWW '22)*. ACM, 2195–2204. https://doi.org/10.1145/3485447.3512092

[32] Ronald J. Williams. 1992. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning* 8, 3-4 (1992), 229–256. https://doi.org/10.1007/BF00992696

[33] Teng Xiao, Suheng Zheng, Zongyu Wang, Yifan Niu, and Suo Feng. 2023. Towards Unbiased and Robust Causal Ranking for Recommender Systems. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (WSDM '23)*. ACM, 402–410. https://doi.org/10.1145/3539597.3570425

[34] Zhiyi Xu, Sibo Wang, Yanyan Shen, and Yuanqing Yu. 2024. Advancing E-Commerce Recommendation with Vision-Language Models. *ACM Transactions on Information Systems* 42, 4 (2024), 1–25. https://doi.org/10.1145/3631525

[35] Zhe Zhao, Lichan Hong, Lighton Wei, Jilin Chen, Aniruddh Nath, Shawn Andrews, Aditee Kumthekar, Maheswaran Sathiamoorthy, Xiyi Yi, and Ed H. Chi. 2019. Recommending What Video to Watch Next: A Multitask Ranking System. In *Proceedings of the 13th ACM Conference on Recommender Systems (RecSys '19)*. ACM, 43–51. https://doi.org/10.1145/3298689.3347045

[36] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Deep Interest Evolution Network for Click-Through Rate Prediction. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI '19)*, Vol. 33. AAAI Press, 5941–5948. https://doi.org/10.1609/aaai.v33i01.33015941

[37] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep Interest Network for Click-Through Rate Prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18)*. ACM, 1059–1068. https://doi.org/10.1145/3219819.3219823

[38] Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. 2024. Uni3D: Exploring Unified 3D Representation at Scale. In *International Conference on Learning Representations (ICLR 2024)*. https://openreview.net/forum?id=wcaE4Dfgt8

[39] Daniel M. Ziegler, Nisheeth Shreshtha, Jeffrey Wu, Daniel Ziegler, Alec Radford, Jeffrey Wu, and Paul F. Christiano. 2019. Fine-Tuning Language Models from Human Preferences. *arXiv preprint arXiv:1909.08593* (2019). arXiv:1909.08593 [cs.CL] https://arxiv.org