Research Square

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

AdaSeq-MTA: Attribution-Aware Adaptive Neural Computation for Multi-Touch Attribution

Jinghui Wang

w.jinghui96@gmail.com

Jiangsu University

Haiyu Liu

Jingxiong Branch, Hebei Expressway Group Co., Ltd.

Liyu Huang

Jiangsu University

Research Article

Keywords: adaptive neural computation, multi-touch attribution, sequence modeling, attribution consistency, knowledge distillation

Posted Date: November 9th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11411594/v1

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

Additional Declarations: No competing interests reported.

AdaSeq-MTA: Attribution-Aware Adaptive Neural Computation for Multi-Touch Attribution

Jinghui Wang[1*], Haiyu Liu[2] and Liyu Huang[1]

[1*]School of Mathematical Sciences, Jiangsu University, 301 Xuefu Road, Zhenjiang, 212013, Jiangsu, China.

[2]Jingxiong Branch, Hebei Expressway Group Co., Ltd., Xiong'an, Hebei, China.

*Corresponding author(s). E-mail(s): wjinghui96@gmail.com;
Contributing authors: 18913113295@163.com; 18770938501@163.com;

Abstract

Processing long customer journeys and estimating touchpoint contributions can make neural multi-touch attribution computationally expensive. Touchpoint selection reduces this cost but may change credit allocations even when conversion predictions remain similar. We propose AdaSeq-MTA, an adaptive neural framework that jointly optimizes conversion prediction, credit consistency, and computational cost. Guided by a frozen full-journey teacher, the model selects a processing path and retention budget for each journey. It maps selected credits back to the original touchpoints and aligns touchpoint and channel allocations, with additional terms for signed contributions and positive-credit mass. A workload-level constraint controls routing, selection, and encoding costs, while confidence-controlled Shapley sampling adjusts attribution effort. On the Criteo benchmark, AdaSeq-MTA retains 66% of touchpoints and reduces inference latency from 18.4 to 13.1 ms per journey, a 28.8% reduction, while the area under the receiver operating characteristic curve (ROC-AUC) changes from 0.7758 to 0.7753. Channel-credit drift decreases from 0.029 for prediction-only adaptive routing to 0.019. With 100 Shapley samples, touchpoint selection reduces attribution latency from 156.4 to 104.2 ms per journey, a 33.4% reduction relative to full-sequence evaluation. These results show that AdaSeq-MTA reduces inference and attribution time with a small change in predictive ROC-AUC, while achieving lower credit drift than prediction-only adaptive routing.

Keywords: adaptive neural computation, multi-touch attribution, sequence modeling, attribution consistency, knowledge distillation

1 Introduction

Users often encounter several advertisements before converting. Multi-touch attribution analyzes these touchpoints and their interactions to allocate conversion credit across touchpoints and channels, supporting marketing analysis. Neural sequence models can capture behavioral dependencies within customer journeys, but processing complete histories and repeatedly evaluating touchpoint contributions increases computational cost. Sequence retrieval, touchpoint selection, and adaptive computation offer practical ways to improve efficiency (Yang et al. 2020; Kumar et al. 2020; Qi et al. 2020; Wang et al. 2018).

Customer journeys differ in channel composition, repeated visits, and temporal spacing as well as length (Fig. 1). Journeys of the same length can contain different amounts of useful information: some touchpoints are highly repetitive, while others reflect important behavioral changes. Efficient multi-touch attribution therefore requires computation to adapt to journey characteristics while retaining information useful for both prediction and credit allocation.

Similar conversion predictions do not guarantee consistent credit allocations. Two models may predict similar conversion probabilities yet respond differently when a touchpoint is removed, leading to different touchpoint or channel credits. Selection changes both the model input and the set of touchpoints participating in attribution, motivating direct constraints on credit changes during compression. Comparing the resulting allocations also requires mapping retained touchpoints back to their original positions. Signed contributions and total credit magnitude complement normalized allocation proportions by capturing additional aspects of credit change.

We propose AdaSeq-MTA, a framework that jointly optimizes conversion prediction, credit consistency, and computational cost. Using a frozen full-journey teacher as a reference, the model draws on channel diversity, repetition, temporal structure, and predictive uncertainty to choose full-sequence processing or order-preserving touchpoint selection and determine a retention budget for each journey. Selected credits are mapped back to the original touchpoints and aligned with the teacher at both touchpoint and channel levels, with signed-marginal and positive-mass terms constraining additional credit changes. A workload-level cost constraint controls overall computation, while a confidence-stopping rule adjusts the number of Shapley samples.

This study focuses on the consistency of model-based credit allocations before and after compression. The teacher provides reference predictions and credits for complete journeys rather than attribution ground truth. Model-based credits differ from the incremental causal effects of advertising exposure, whose identification requires appropriate data and assumptions (Singal et al. 2019; Yao et al. 2022). The experiments separately examine teacher–student credit discrepancies and routing-induced credit drift between full and selected paths within the same student model.

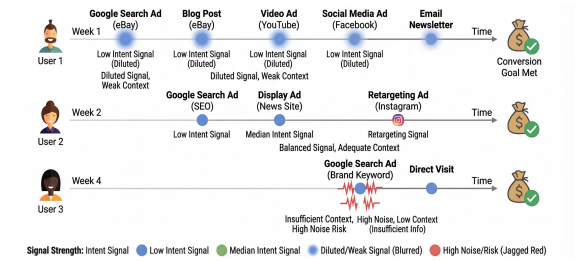


Fig. 1 Illustrative customer journeys with different touchpoint patterns and contextual evidence.

Our main contributions are:

- **Joint optimization of prediction, credit, and cost.** We optimize conversion prediction and credit consistency under an overall computation budget, making the attribution effects of touchpoint compression part of resource allocation.
- **Credit-consistency objectives at touchpoint and channel levels.** We map selected credits back to original touchpoints and combine distribution alignment with signed-marginal and positive-mass terms. Staged LOO and Shapley supervision connects training objectives to the final attribution computation.
- **Adaptive computation based on journey characteristics.** We use channel, temporal, and uncertainty features to select processing paths and retention budgets, together with a separate confidence-stopping rule for attribution sampling, combining touchpoint selection with sampling-precision control.

Routing-policy comparisons, credit-objective ablations, backbone comparisons, and efficiency analyses evaluate the trade-offs among prediction performance, credit consistency, and computational cost.

2 Related Work

2.1 Neural Prediction over Customer Journeys

Customer journeys connect heterogeneous digital touchpoints to marketing outcomes and managerial actions. AI-enabled journey-management research frames intelligent tools as supporting the analysis, design, engagement, and guidance of customer interactions (Rusthollkarhu et al. 2022). In digital attribution, intelligent systems have combined real-time conversion prediction with channel-level decision support, including Bayesian-network models designed for imbalanced e-commerce journeys (Ben Mrad and Hnich 2024). These studies motivate treating MTA as an auditable marketing decision-support problem rather than only a sequence-classification benchmark.

CTR and CVR prediction provide the predictive foundation for such systems. Reviews of online-advertising prediction organize the field around statistical, factorization, deep, and tree-based models (Yang and Zhai 2022), while recent work surveys knowledge-aware CTR models and their treatment of sparse and cold-start interactions (Jin and Yang 2025). Representative architectures include DeepFM (Guo et al. 2017b), DIN and DIEN (Zhou et al. 2018b,a), and multi-task recommendation models such as MMoE and PLE (Ma et al. 2018; Tang et al. 2020). Delayed-feedback correction and calibration remain important because a ranking gain does not guarantee reliable conversion probabilities (Cao et al. 2022).

Long-history systems such as SIM, UBR, and MIMN retrieve or maintain target-relevant user interests to control the cost of extensive behavioral histories (Qi et al. 2020; Chen et al. 2021; Pi et al. 2019). Their retrieval mechanisms motivate our long-sequence coarse-to-fine module (LSCM), but Top-K selection is not claimed as new. Existing customer-journey predictors generally do not test whether computational compression preserves the resulting touchpoint and channel conclusions.

2.2 Multi-Touch Attribution and Credit Fidelity

Rule-based attribution assigns credit through fixed conventions such as first-click, last-click, and equal allocation (Jordan et al. 2011; Abhishek et al. 2012). Neural approaches, including DeepMTA and attention-based attribution, learn representations of observed journeys (Yang et al. 2020; Kumar et al. 2020). CausalMTA introduces explicit causal modeling assumptions (Yao et al. 2022), while industrial attribution systems connect individual journeys with aggregate marketing analysis (Bencina et al. 2025). These paradigms use different credit-allocation rules (Fig. 2) and differ in the meaning of their outputs.

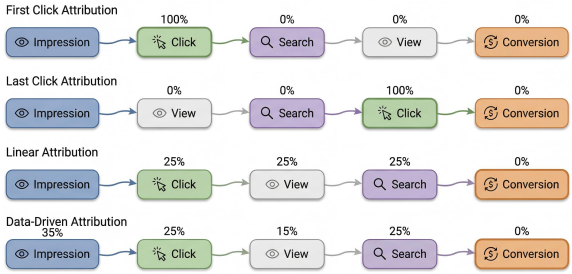


Fig. 2 Schematic comparison of credit allocation across multi-touch attribution paradigms.

Credit fidelity in this study concerns a specific transformation: replacing full-journey computation with a learned selected path. The full and selected predictors induce different coalition games, so their credit vectors require an explicit index mapping before comparison. Teacher agreement measures the combined effect of parameter adaptation and processing-path changes; external attribution targets provide a separate assessment of the resulting credit quality. Multi-attribution learning and benchmark work motivate evaluating against defined target quantities rather than inferring attribution correctness from conversion accuracy (Chen et al. 2025; Wu et al. 2026).

Post-hoc explanations also differ in their reference quantities. LIME fits local surrogate models, whereas SHAP expresses local importance through additive Shapley-based explanations (Ribeiro et al. 2016; Lundberg and Lee 2017). Integrated Gradients uses a path-based attribution rule, and DeepLIFT propagates activation differences relative to a reference input (Sundararajan et al. 2017; Shrikumar et al. 2017). These constructions should not be treated

as interchangeable credit targets. At the population level, SAGE evaluates feature contributions to predictive power (Covert et al. 2020); this differs from distributing a particular journey's conversion credit.

Explanation quality also requires checks beyond prediction accuracy. Attention weights need not faithfully explain model predictions (Jain and Wallace 2019), and saliency-map randomization tests can reveal dependence on learned parameters and data (Adebayo et al. 2018). Robustness and infidelity criteria examine complementary properties of an explanation (Alvarez-Melis and Jaakkola 2018; Yeh et al. 2019). These findings motivate separating credit stability under compression from external credit validity in our evaluation.

2.3 Adaptive Neural Computation and Attribution Consistency

Efficient recommender systems must balance predictive effectiveness with deployment cost. Knowledge distillation can transfer a high-capacity teacher into a scalable student (Jose and Shetty 2022); related studies also use multiple teachers to address recommendation bias rather than treating all teacher signals as equally reliable (Yang et al. 2025). AdaSeq-MTA similarly uses a frozen teacher, but transfers prediction and mapped credit structure rather than only logits or representations. This internal target can preserve a teacher's error, so external-credit evaluation remains separate.

Classical distillation transfers softened outputs (Hinton et al. 2015), while FitNets transfers intermediate hints and attention transfer aligns teacher and student attention maps (Romero et al. 2015; Zagoruyko and Komodakis 2017). Relational and similarity-preserving distillation extend the target to relationships among examples (Park et al. 2019; Tung and Mori 2019). These precedents establish that output matching alone is not the only form of knowledge transfer. Our target instead compares signed and normalized coalition contributions after restoring the original journey indices.

Adaptive Computation Time and PonderNet learn how many computation steps to allocate (Graves 2016; Banino et al. 2021). FastBERT combines adaptive inference with self-distillation (Liu et al. 2020). These approaches motivate input-dependent processing and the need to measure the resulting accuracy–cost trade-off.

Conditional-computation methods allocate different workloads to different inputs or tokens. SkipNet learns input-dependent layer skipping (Wang et al. 2018); DynamicViT removes less informative tokens (Rao et al. 2021); TokenLearner constructs a compact adaptive token set (Ryoo et al. 2021); and Mixture-of-Depths routes a capacity-limited subset of sequence positions (Raposo et al. 2024). These works provide technical precedents for dynamic routing, but primarily optimize prediction–efficiency trade-offs. They do not require a marketing-credit distribution computed after routing to remain aligned with its full-input counterpart.

AD-KD transfers token-level Integrated Gradients and multi-view attribution targets from a teacher to a student (Wu et al. 2023). Attribution transfer itself is therefore an established compression strategy. AdaSeq-MTA focuses on the additional change in the player set caused by evidence selection: it restores original touchpoint indices, aligns channel aggregates and signed credit mass, and learns the retained evidence under a workload constraint. This distinction concerns the joint formulation, not a claim to introduce attribution distillation.

Explanation-guided learning already constrains model behavior through explanation penalties, including input-gradient regularization (Ross et al. 2017). AdaSeq-MTA addresses the narrower problem of aligning coalition-derived credits between full and selected chronological journeys while learning their processing budget. Its distinction lies in this joint formulation and the sign- and scale-sensitive mapped objective, rather than in the general idea of explanation supervision.

The proposed formulation differs from prediction-oriented routing and efficient sequence encoding (Wang et al. 2018; Rao et al. 2021; Ryoo et al. 2021; Raposo et al. 2024) in its optimization target. A prediction-only router may discard observed touchpoints while preserving AUC yet materially change channel credit.

AdaSeq-MTA therefore treats the full-observed-journey touchpoint/channel distribution as an internal teacher target and optimizes mapped-credit distortion jointly with the learned routing workload. This target measures teacher–routed discrepancy, including student adaptation, not attribution truth. Coalition sample count is not learned jointly: it is controlled afterward by a separate statistical stopping rule, and encoder versus attribution costs are reported as two distinct axes. Multi-attribution learning and MAC provide complementary label views and an external evaluation setting (Chen et al. 2025; Wu et al. 2026); they do not by themselves decide how much observed evidence a journey should retain. Prediction-only routing provides a system-level comparison; attributing a difference solely to credit supervision additionally requires identical deployed architecture and training conditions.

3 Methods

The method couples two learning decisions: which observed evidence to retain and how closely the resulting predictor should preserve full-journey credit. The router and retention budget define the evidence path; Eq. (17) measures its credit distortion; and Eq. (21) couples that distortion and prediction loss to workload cost. The backbone supplies the predictive value function, while Sec. 3.5 controls attribution estimation precision after the path is selected.

Figure 3 summarizes the training, adaptive prediction, and fixed-route attribution pipeline.

3.1 Problem Formulation and Credit Scope

Multi-touch attribution (MTA) in online advertising aims to allocate model-based conversion credit across the observed touchpoints of a customer journey. Given a dataset $\mathcal{D} = \{(u_i, J_i, y_i)\}_{i=1}^{N}$, $u_i \in \mathbb{R}^{d_u}$ contains prediction-time context, $J_i = (p_{i1}, ..., p_{i,T_i})$ is a chronologically ordered sequence of variable length T_i, each touchpoint $p_{it} = (c_{it}, f_{it})$ contains channel identifier c_{it} and features $f_{it} \in \mathbb{R}^{d_f}$ available at or before the event timestamp t_{it}, and $y_i \in \{0, 1\}$ is the conversion label. The accounting output assigns $\alpha_{it} \in [0, 1]$ to every observed touchpoint and satisfies

$$\sum_{t=1}^{T_i} \alpha_{it} = y_i, \quad \forall i \tag{1}$$

This conservation rule is an accounting convention, not a causal estimand. Let $\hat{r}_{it}$ be the marginal change in the trained predictor under the order-preserving coalition definition in Sec. 3.5. Final credits use one definition throughout the paper:

$$\alpha_{it} = y_i \frac{\max(0, \hat{r}_{it})}{\sum_{t'=1}^{T_i} \max(0, \hat{r}_{it'})}. \tag{2}$$

Equation (2) applies when at least one retained score is positive. When all rectified scores are zero, the label-independent normalized distribution is uniform over retained touchpoints. Multiplication by the observed label gives this allocation to converted journeys and zero accounting credit to non-converted journeys. Touchpoints excluded by LSCM always receive exactly zero before and after normalization. Conservation is exact by construction, but the resulting α_{it} remains model-based credit and is not identified as an individual treatment effect.

Beyond confounding, MTA faces heterogeneous evidence sufficiency and cost. Length T_i is observable but incomplete: journeys of equal length can differ in channel diversity, repetition, temporal spacing, and predictive uncertainty. We therefore use short/medium/long groups only for descriptive auditing. The operational policy is learned in Sec. 3.2 and is compared with a length-only router in the experiments.

To address this, we condition computation on observed evidence sufficiency and uncertainty rather than sequence length alone. We define a budget-aware attribution function:

$$(\hat{r}_{it}, \beta_{it}^{R}) = g_\psi(t \mid J_i, u_i), \qquad \alpha_{it} = y_i \beta_{it}^{R}. \tag{3}$$

Here g_ψ denotes the complete routing and attribution procedure with all fitted routing and student parameters ψ, not an additional prediction head. At prediction time it produces signed contributions $\hat{r}_{it}$ and their rectified, normalized distribution β_{it}^{R} on the original index space, with the retained-position uniform fallback defined above. Neither output requires the realized conversion label. Only after the outcome window has matured

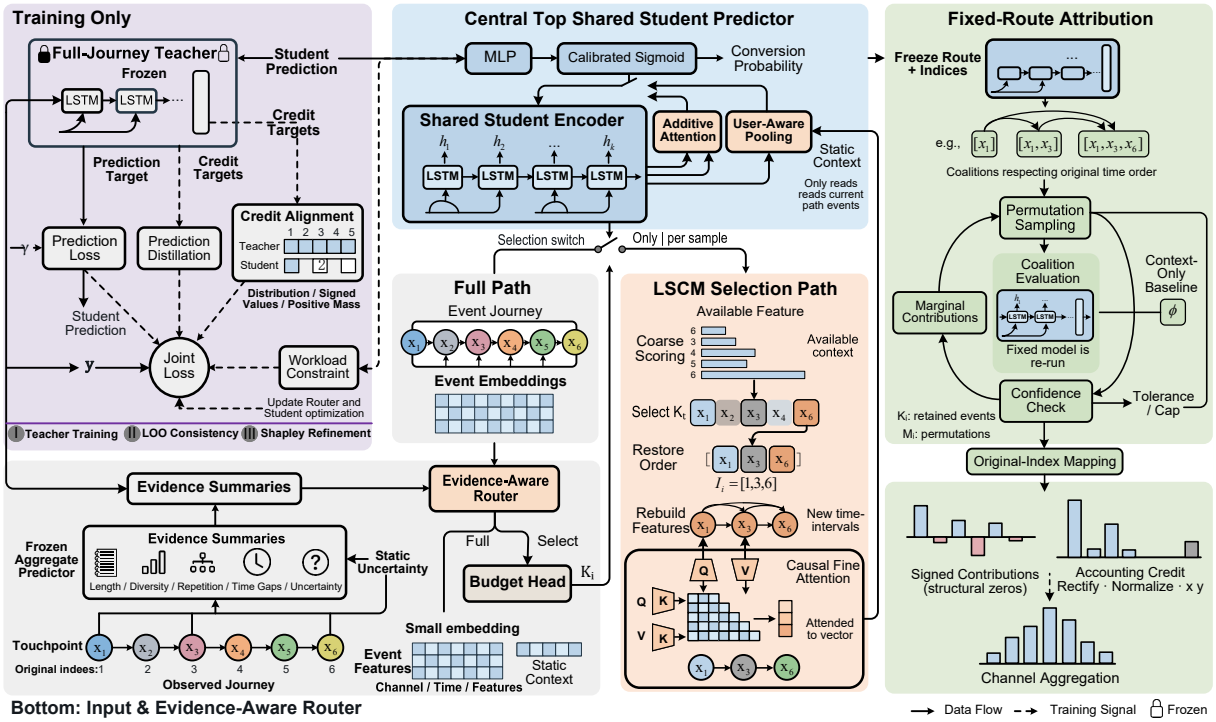


Fig. 3 Overall architecture of AdaSeq-MTA, integrating adaptive routing, teacher-guided learning, and Shapley attribution.

is y_i used to form the accounting allocation α_{it}. Thus an unobserved outcome is not treated as zero, and the label is never an input to routing, selection, or coalition prediction. The subsequent sections define the adaptive policy and its feasible expert operations.

3.2 Evidence-Aware Routing, Consistency, and Budget Control

Fixed length thresholds are retained only as descriptive strata for evaluation. The operational decision is learned from a journey evidence vector

$$\mathbf{g}_i = \big[\log(1+T_i), H_c(J_i), R_{\text{dup}}(J_i), \\ \text{mean}(\Delta t_i), \text{std}(\Delta t_i), U_i\big], \tag{4}$$

where H_c is channel entropy and R_{dup} is the repeated-channel occurrence fraction defined below, and $U_i = -p_i^{(0)} \log p_i^{(0)} - (1-p_i^{(0)}) \log(1-p_i^{(0)})$ is predictive entropy from an aggregate-only logistic model trained on the training partition and frozen before router training. That model consumes static request/user context and the same count, entropy, repetition, and time-gap summaries in $\mathbf{g}_i$; it does not run the LSTM, LSCM, attention, or any full-journey embedding encoder. All non-uncertainty summaries are updated in one streaming pass over logged touchpoints and are available at the prediction timestamp.

Boundary cases and repeated-channel counts. A model input journey contains at least one observed event. Records with $T_i = 0$ are excluded before forming this study's prediction and attribution datasets; no zero-event attribution is produced. The empty coalition in Sec. 3.5 is a model evaluation baseline, not an additional observed journey. Let $n_{ic} = \sum_{t=1}^{T_i} \mathbb{I}[c_{it} = c]$ and let C_i^{obs} count distinct channel categories in the full observed journey. We use

$$\begin{aligned} R_{\text{dup}}(J_i) &= \frac{\sum_c \max(n_{ic}-1, 0)}{T_i} \\ &= 1 - \frac{C_i^{\text{obs}}}{T_i}, \\ H_c(J_i) &= -\sum_{c:n_{ic}>0} \frac{n_{ic}}{T_i} \log \frac{n_{ic}}{T_i}. \end{aligned} \tag{5}$$

All occurrences after the first in each channel are counted, whether consecutive or separated by other channels. The definition uses channel identity only, not ad, product, or complete-event equality. Missing channel IDs use a dedicated missing category under the fixed preprocessing dictionary.

Let the interval list be $D_i = (t_{i2} - t_{i1}, \ldots, t_{iT_i} - t_{i,T_i-1})$ in a fixed time unit specified as part of the preprocessing configuration. For $n_i = T_i - 1 > 0$, define $\overline{D}_i = n_i^{-1} \sum_{k=1}^{n_i} D_{ik}$ and population standard deviation $s_i = \sqrt{n_i^{-1} \sum_k (D_{ik} - \overline{D}_i)^2}$. For $n_i = 0$, both statistics are zero by definition. Thus one interval has zero standard deviation, and a one-touch journey has $H_c = R_{\text{dup}} = \overline{D}_i = s_i = 0$. No additional no-interval flag is used: $\log(1+T_i)$ already distinguishes the singleton case from multi-event journeys with tied timestamps. Events are stably ordered by timestamp and original event index; tied timestamps yield zero gaps. Records with missing or invalid timestamps are excluded under a fixed preprocessing rule rather than given fabricated gaps. These conventions are applied before evaluating the evidence vector or uncertainty predictor. For $T_i = 1$, only the full route is feasible, with probability one, and the budget head is not evaluated.

The evidence-aware router (EER) assigns one of two core paths,

$$\boldsymbol{\pi}_i = \text{softmax}(\text{MLP}_\phi(\mathbf{g}_i)), \qquad e_i \in \{\text{full}, \text{select}\}. \tag{6}$$

The full path uses every observed touchpoint; select uses the index-preserving LSCM. Selection is infeasible when no candidate budget is smaller than T_i. During a router warm-up, feasible paths and budgets are sampled with a nonzero probability floor; training then samples discrete paths and budgets from their masked categorical probabilities. Continuous categorical relaxations are established alternatives (Jang et al. 2017; Maddison et al. 2017); here a likelihood-ratio estimator supplies the gradients of hard structural choices. Validation and test use deterministic $\arg\max_e \pi_{ie}$. The initial warm-up lasts exactly E_{warm} epochs and follows the prespecified exploration schedule.

For the selection expert, a second router assigns a journey-specific retained budget from a fixed set of distinct positive integers $\mathcal{K} = \{K_1, \ldots, K_{|\mathcal{K}|}\}$:

$$\begin{aligned} \boldsymbol{\rho}_i &= \text{softmax}(\text{MLP}_K([\mathbf{g}_i; \boldsymbol{\pi}_i])), \\ b_i^{\text{hard}} &= \arg\max_b \rho_{ib}, \\ K_i^{\text{hard}} &= \min(T_i, K_{b_i^{\text{hard}}}), \\ \widetilde{K}_i &= \min\left(T_i, \sum_{b=1}^{|\mathcal{K}|} \rho_{ib} K_b\right). \end{aligned} \tag{7}$$

During training, $e_i \sim \widetilde{\boldsymbol{\pi}}_i$ and $b_i \sim \widetilde{\boldsymbol{\rho}}_i$ for the selection path; the argmax equations specify deterministic evaluation. Infeasible budgets are masked before normalization. $\widetilde{K}_i$ describes the expected budget under the unmixed conditional distribution $\boldsymbol{\rho}_i$; the actual training expectation uses $\widetilde{\boldsymbol{\rho}}_i$. Neither quantity differentiates a hard sequence slice. Costs and credit losses are evaluated on the realized hard path. Shapley sample counts are controlled separately by Sec. 3.5.

Let $\mathcal{E}_i$ and $\mathcal{B}_i$ be the feasible route and budget sets. During training epoch t, the scheduled exploration coefficient $\eta_t \in [0, 1)$ is fixed independently of the current action. The actual distributions are

$$\begin{aligned} \widetilde{\pi}_{ie} &= (1 - \eta_t)\pi_{ie} + \eta_t / |\mathcal{E}_i|, \\ \widetilde{\rho}_{ib} &= (1 - \eta_t)\rho_{ib} + \eta_t / |\mathcal{B}_i|. \end{aligned} \tag{8}$$

These formulas apply only to feasible choices; all others have probability zero. The budget distribution is used only when selection is feasible and selected. Warm-up uses positive η_t, followed by a prespecified decay; evaluation uses the unmixed argmax distributions. The likelihood-ratio estimator differentiates the logarithm of the mixed probabilities, not that of the unmixed softmax.

Gradients for discrete processing decisions. Let $a_i = (e_i, b_i, o_i)$ include the route, budget, and an ordered sample $o_i = (o_{i1}, \ldots, o_{iK_i})$ of distinct touchpoints. Given coarse scores ζ_{it} and fixed positive selection temperature τ_{sel}, selection samples without replacement with

$$Q_i(o_i \mid b_i) = \prod_{k=1}^{K_i} \frac{\exp(\zeta_{i,o_{ik}}/\tau_{\text{sel}})}{\sum_{t \notin \{o_{i1}, \ldots, o_{i,k-1}\}} \exp(\zeta_{it}/\tau_{\text{sel}})}. \tag{9}$$

The sampled indices are then sorted chronologically to form I_i; sampling order is never passed to the encoder. For selection, $P_i(a_i) = \widetilde{\pi}_{i,\text{select}} \widetilde{\rho}_{ib_i} Q_i(o_i \mid b_i)$; for the full path, $P_i(a_i) =$

$\widetilde{\pi}_{i,\text{full}}$. These are the actual sampling probabilities, including any warm-up mixture. Let $\ell_i(a_i)$ be the per-journey prediction, distillation, and credit objective, and ψ collect all trainable parameters. With $\mu_F = N^{-1}\sum_i \mathcal{C}_i^{\text{full}}$ fixed, define

$$\begin{aligned} f_i(a_i) &= \ell_i(a_i) + \lambda_B \mathcal{C}_i(a_i)/\mu_F, \\ \nabla_\psi \mathbb{E}_{a_i}[f_i] &= \mathbb{E}_{a_i}\big[\nabla_\psi f_i\big|_{a_i}\big] \\ &\quad + \mathbb{E}_{a_i}\big[(f_i - \nu_i)\nabla_\psi \log P_i(a_i)\big]. \end{aligned} \tag{10}$$

Here ν_i is an action-independent baseline determined before drawing a_i, and the multiplier $f_i - \nu_i$ is detached in the likelihood-ratio term. The first term backpropagates through the realized coalition predictions with indices fixed; the second trains route, budget, and coarse-score choices from their prediction, credit, and cost consequences. No gradient is taken through sorting, integer indices, or the zero-filled excluded positions themselves. The estimator targets the expected stochastic-policy objective and can have high variance. Validation and deployment use deterministic argmax and hard Top-K.

Staged attribution-consistency teacher. Training proceeds in three prespecified stages. Stage I trains a full-observed-journey teacher with binary cross-entropy. A scalar temperature τ_F is fitted on validation teacher logits and frozen for every nonempty teacher input. Empty inputs use the separately calibrated context-only baseline in Sec. 3.5. All teacher modules and both calibration parameters are then frozen. Stage II trains the router and routed student using cached calibrated teacher predictions and non-negative leave-one-out (LOO) scores

$$\begin{aligned} r_{it}^F &= p_i^F(J_i) - p_i^F(J_i \setminus p_{it}), \\ b_{it}^F &= \max\{0, r_{it}^F\}, \\ \boldsymbol{\beta}_i^F &= \text{Normalize}(\mathbf{b}_i^F). \end{aligned} \tag{11}$$

with retained events restored to chronological order. For both LOO and Shapley evaluations, event-local attributes and absolute timestamps remain fixed, while event-relative features (including the previous-touch time gap) are recomputed from the retained chronological sequence using the same fixed preprocessing and start-of-journey convention. The route and selected index set remain fixed as specified in Sec. 3.5. Stage III replaces LOO targets on a fixed training subset with low-sample Shapley targets from the frozen teacher, evaluating coalitions in chronological order, and fine-tunes the router/student at a lower learning rate. This aligns the training target with final evaluation while retaining LOO as a documented warm-start proxy. The objective ablation compares LOO-only training, LOO continuation, and sampled-Shapley refinement. The Stage-III target is

$$\begin{aligned} \boldsymbol{\beta}_i^{F,SV} = \text{Normalize}\Big(\\ \Big[\max\{0, \widehat{SV}_{i,t}^F(M_{\text{train}})\}\Big]_{t=1}^{T_i}\Big). \end{aligned} \tag{12}$$

where M_{train} is fixed on validation data and is smaller than the maximum evaluation sample count.

Separating adaptation and routing effects. Let $\boldsymbol{\varphi}_i^F$ be the full-teacher signed Shapley vector, $\boldsymbol{\varphi}_i^{S,\text{full}}$ the same journey evaluated with the fitted student's full path, and $\boldsymbol{\varphi}_i^R$ its realized routed-path vector mapped to the original indices. The two student games use identical fitted parameters, context, empty-coalition baseline, and student temperature; forcing the full path changes only the processing path. Then

$$\begin{aligned} \boldsymbol{\varphi}_i^R - \boldsymbol{\varphi}_i^F = (\boldsymbol{\varphi}_i^R - \boldsymbol{\varphi}_i^{S,\text{full}}) \\ + (\boldsymbol{\varphi}_i^{S,\text{full}} - \boldsymbol{\varphi}_i^F). \end{aligned} \tag{13}$$

The first term isolates the selected processing path, including its refinement operation; the second captures student adaptation and any teacher–student calibration difference. The identity concerns vectors, not additive JS divergences or additive error norms. Comparing these three discrepancies requires identical journeys and common sampled permutations restricted to each player set. A teacher–routed loss remains the optimization target; it is not interpreted as a pure measure of touchpoint deletion.

The routed credit is computed from the same attribution operator as the stage target. Stage-II examples are drawn independently with replacement from a fixed mixture of nonempty length/outcome strata. The known per-example probability $q(i) > 0$ includes stratum selection and within-stratum sampling. Converted strata may be oversampled, but batches are not rejected

or resampled when they contain no conversion; such a batch has a zero estimated credit term while prediction and distillation remain active. Known sampling probabilities are used to importance-weight prediction, distillation, credit, and cost terms to their specified full-partition objectives. In Stage II, the routed student uses chronological LOO marginals; in Stage III, it uses student Shapley marginals evaluated with the same sampled permutations as the frozen teacher (common random numbers). The following marginal formulas apply only to retained positions $t \in I_i$; define $r_{it}^R = 0$ separately for every $t \notin I_i$ before forming the full-length vectors:

$$\begin{aligned} r_{it}^R &= p_i^R(J_i^R) \\ &\qquad - p_i^R(J_i^R \setminus p_{it}) \quad \text{(Stage II)}, \\ r_{it}^R &= \widehat{SV}_{i,t}^R(M_{\text{train}}) \quad \text{(Stage III)}, \\ b_{it}^R &= \max\{0, r_{it}^R\}, \\ \boldsymbol{\beta}_i^R &= \text{Normalize}([b_{it}^R]_{t=1}^{T_i}). \end{aligned} \tag{14}$$

This zero extension does not evaluate a Shapley value for a non-player. The rectified and normalized vectors above include all original positions, with excluded positions explicitly zero before normalization. The fixed Stage-III refinement subset is selected once from the training partition by prespecified strata of journey length and channel diversity. Frozen-teacher Shapley targets and their permutation identifiers are cached once. Student coalition predictions use the two-pass recomputation procedure below with fixed common permutations; a single nonlinear credit loss is formed from the complete marginal vector. No test journey or test-derived threshold enters subset selection.

Stage-III objective and sampling. Let $\mathcal{S}_+ \subseteq \mathcal{D}_+$ be the fixed, nonempty subset with cached Shapley targets. Refinement uses two independent sample streams. The first samples the full training partition with $q_D(i) > 0$ and retains prediction, distillation, and cost terms with the original N, D_F. The second samples $\mathcal{S}_+$ with $q_S(i) > 0$ and estimates the explicitly subset-defined objective

$$\begin{aligned} \mathcal{L}_{\text{credit}}^{III} &= \frac{1}{|\mathcal{S}_+|} \sum_{i \in \mathcal{S}_+} \mathbb{E}_{a_i}[L_i^{SV}(a_i)], \\ \widehat{\mathcal{L}}_{\text{credit}}^{III} &= \frac{1}{m_S} \sum_{k=1}^{m_S} \frac{L_{i_k}^{SV}(a_{i_k})}{|\mathcal{S}_+| q_S(i_k)}. \end{aligned} \tag{15}$$

Here L_i^{SV} is the complete per-journey bracket of the mapped-credit objective with sampled Shapley inputs. It replaces the Stage-II LOO credit term; it is not claimed to estimate a full-population Shapley objective. The two streams' weighted gradients are added before each update. The full-data stream uses prediction, distillation, and budget returns in the direct and likelihood-ratio estimator; the subset stream uses $\lambda_{\text{attr}} L_i^{SV}$ with its own action-independent baseline and subset weights. Prediction and cost are not counted again on the subset stream. Every Stage-III epoch retains the full-partition stochastic budget update and deterministic validation checks. An empty $\mathcal{S}_+$ disables this refinement stage.

Exact gradient of the sampled nonlinear loss. For a fixed sampled action and fixed M_{train} permutations, write the zero-mapped marginal vector as $\mathbf{r}(\theta) = \sum_{b=1}^{B_c} \mathbf{r}_b(\theta)$, where each block already includes its $1/M_{\text{train}}$ factor. In pass one, compute and accumulate the full vector without retaining coalition computation graphs. Form the entire normalized, signed, and mass-sensitive loss $L_i^{SV}(\mathbf{r})$ once and obtain $\mathbf{g} = \partial L_i^{SV}/\partial \mathbf{r}$ by treating the accumulated vector as a differentiable leaf. In pass two, recompute each block with gradients enabled and backpropagate the scalar

$$\text{stopgrad}(\mathbf{g})^\top \mathbf{r}_b(\theta). \tag{16}$$

Summing these block gradients gives $(\partial \mathbf{r}/\partial \theta)^\top \mathbf{g}$, the gradient of the single complete sampled loss wherever differentiable, rather than an average of blockwise losses. Both passes use the same branch-derivative convention. Rectification uses derivative zero at zero, and the uniform-fallback distribution is treated as constant on its active branch. At the discontinuous zero-mass boundary this is an implementation convention, not a claim of a classical derivative; signed and mass terms retain their own branch derivatives. Parameters, route, selected indices, coalition features, and permutation IDs remain unchanged between passes; coalition evaluation uses deterministic layers with dropout disabled and no mutable running-statistic updates. The first-pass scalar loss is detached in the action likelihood-ratio term and used once per sampled action, with $\log P_i(a_i)$ recomputed at the unchanged parameters. The optimizer steps only after both passes and both data streams finish. Memory scales with one coalition block plus the

marginal vector and shared model state; the extra recomputation and direct prediction passes are included in reported training cost. Away from the stated nondifferentiable boundaries, this exactly differentiates the fixed Monte Carlo surrogate; it is not an unbiased gradient claim for the nonlinear exact-Shapley loss.

If all rectified scores of a game vanish, its fallback is uniform over that game's players: all original touchpoints for the full teacher and only retained touchpoints for the routed student. Student-excluded positions remain zero. For the loss only, add $\varepsilon_{\mathrm{JS}} > 0$ per coordinate and renormalize on the common original support. Channel vectors are aggregated from the unsmoothed touchpoint distribution first, then smoothed on the same observed-channel support for both models. Reported credits remain unsmoothed with exact excluded-position zeros; reported JS uses the continuous convention $0 \log(0/q) = 0$. The numerical floor is fixed before evaluation.

Channel-level distributions $\mathbf{a}_i^F$ and $\mathbf{a}_i^R$ are obtained by summing normalized non-negative credits by channel. To penalize magnitude changes hidden by normalization, define positive masses $q_i^F = \sum_t b_{it}^F$ and $q_i^R = \sum_t b_{it}^R$. Let $\mathcal{D}_+ = \{i : y_i = 1\}$; mapped-credit consistency is defined only for converted journeys:

$$\begin{aligned}\mathcal{L}_{\mathrm{attr}} = \frac{1}{|\mathcal{D}_+|} \sum_{i \in \mathcal{D}_+} \Big[& D_{\mathrm{JS}}(\boldsymbol{\beta}_i^F \| \boldsymbol{\beta}_i^R) \\ & + \gamma_c D_{\mathrm{JS}}(\mathbf{a}_i^F \| \mathbf{a}_i^R) \\ & + \frac{\gamma_s}{T_i} \sum_{t=1}^{T_i} \mathrm{Huber}_\kappa(r_{it}^F - r_{it}^R) \\ & + \gamma_m \big| \log(\epsilon_m + q_i^F) \\ & \qquad - \log(\epsilon_m + q_i^R) \big| \Big].\end{aligned} \tag{17}$$

where r_{it}^F denotes LOO marginals in Stage II and sampled teacher Shapley values in Stage III; $\boldsymbol{\beta}_i^F$ uses their rectified normalization. We use $\mathrm{Huber}_\kappa(x) = x^2/2$ for $|x| \le \kappa$ and $\kappa(|x| - \kappa/2)$ otherwise, with $\kappa > 0$; no division by κ is applied. Here $\epsilon_m > 0$, and both mass and JS terms use natural logarithms. The coefficients $\gamma_c, \gamma_s, \gamma_m, \lambda_{\mathrm{KD}}, \lambda_{\mathrm{attr}}$ are non-negative. Numerical floors are not fitted on test credits. The signed Huber term detects suppressive-touch and scale changes, while the log-mass term penalizes proportional shrinkage that normalization would hide. Teacher scores with zero positive mass use the same uniform fallback as Eq. (2), but their zero mass remains visible to the magnitude term. Non-converted journeys contribute prediction and distillation losses but no normalized conversion-credit target. Teacher calibration is performed once in Stage I and is not optimized through the student loss. Signed marginals, positive mass, total game effect $v_i(A_i) - v_i(\varnothing)$, and non-negative accounting credit are distinct quantities.

Why distribution alignment is insufficient. Consider positive teacher marginals $(0.8, 0.2)$ and student marginals $(0.08, 0.02)$. Their normalized credit distributions coincide, although their positive masses differ by a factor of ten. Distribution divergence alone is therefore zero in this example; the mass term in Eq. (17) detects the change. Likewise, rectification removes negative marginals from the normalized target, motivating the separate signed penalty.

For the recurrent core, the fixed-coefficient deployment routing-cost proxy includes the base uncertainty predictor, router, full scan and partial sorting required by selection, retained sequence encoding, and the quadratic fine-attention term:

$$\begin{aligned}\mathcal{C}_i^{\mathrm{route}} &= c_0 + c_r + c_f T_i + c_e K_i^{\mathrm{eff}} \\ &\quad + \mathbb{I}[e_i = \mathrm{select}] \big[c_p T_i + c_s T_i \log(1 + K_i) \\ &\qquad + c_q K_i + c_a K_i^2 \big], \\ \mathcal{C}_i^{\mathrm{full}} &= c_0 + c_r + c_f T_i + c_e T_i.\end{aligned} \tag{18}$$

where $K_i^{\mathrm{eff}} = T_i$ for the full path and equals the realized hard budget for the selection path. Here $c_f T_i$ accounts for shared feature/summary extraction, $c_p T_i$ for selection-only coarse projections, and $c_q K_i$ for fine-attention input/output projections at fixed widths. The full-path denominator includes the same uncertainty predictor and router overhead to isolate the path saving. This denominator therefore differs from a teacher-only system without routing overhead. All coefficients are fixed from operation accounting or a training-partition hardware calibration, never fitted to test latency. This is the deployment-path cost objective even when evaluated on sampled training actions; without-replacement sampling and backpropagation add training overhead outside this deployment proxy. Kernel fusion and batching

can change the relation between this analytical proxy and wall-clock latency. The proxy defines the recurrent-core optimization constraint; the latency tables provide separate runtime comparisons. A Transformer implementation requires its own full-path attention cost model; its measured comparison does not validate the recurrent proxy for that architecture. Deployment attribution cost is reported separately because its M_i is set by statistical precision rather than the learned router.

We constrain the ratio of total routed workload to total full-path workload, rather than averaging per-journey ratios that would give a one-touch journey and a very long journey equal weight. Let

$$R_{\mathcal{D}} = \frac{\sum_{i\in\mathcal{D}} \mathcal{C}_i^{\text{route}}}{\sum_{i\in\mathcal{D}} \mathcal{C}_i^{\text{full}}}. \tag{19}$$

Before training, an optimistic cost-only lower bound is computed separately on the training and validation workloads,

$$B_{\min} = \frac{\sum_i \min\big(\mathcal{C}_i^{\text{full}}, \min_{K_b<T_i} \mathcal{C}_i^{\text{select}}(K_b)\big)}{\sum_i \mathcal{C}_i^{\text{full}}}, \tag{20}$$

where an empty inner feasible set leaves the full-path cost. This oracle bound ignores policy expressivity, exploration floors, and quality constraints, so satisfying it is necessary but not sufficient for an acceptable operating point. A requested $B < B_{\min}$ is rejected before optimization; it is never made feasible by silently dropping journeys or accuracy terms. We impose $R_{\mathcal{D}} \le B$ with a Lagrangian rather than a fixed soft-penalty weight:

$$\begin{aligned}\mathcal{L}_{\text{core}} &= \mathcal{L}_{\text{pred}} + \lambda_{\text{KD}}\mathcal{L}_{\text{KD}} \\ &\quad + \lambda_{\text{attr}}\mathcal{L}_{\text{attr}} + \lambda_B(R_{\mathcal{D}} - B), \qquad (21)\\ \lambda_B &\leftarrow \max\left\{0, \lambda_B + \eta_B\left(R_{\mathcal{D}} - B\right)\right\}. \qquad (22)\end{aligned}$$

The training denominator $D_F = \sum_{i=1}^{N} \mathcal{C}_i^{\text{full}}$ is computed once and held fixed. For independent draws $i_1, \ldots, i_m$ from this length/outcome-stratified sampler with known probabilities $q(i) > 0$, the budget estimator is

$$\widehat{R}_{\mathcal{B}} = \frac{1}{mD_F} \sum_{k=1}^{m} \frac{\mathcal{C}_{i_k}(a_{i_k})}{q(i_k)}. \tag{23}$$

Consequently $\mathbb{E}[\widehat{R}_{\mathcal{B}}] = D_F^{-1} \sum_i \mathbb{E}_{a_i} \mathcal{C}_i(a_i)$. A random batch denominator is not used. Prediction and distillation terms use weights $1/[Nq(i)]$; converted-only credit terms use $\mathbb{I}[y_i = 1]/[N_+ q(i)]$, with the full training counts N, N_+ fixed. Equivalently, ℓ_i in Eq. (10) includes $(N/N_+)\mathbb{I}[y_i = 1]$ on its credit term. The same sampling weights apply to both direct and likelihood-ratio gradients. If $N_+ = 0$, the credit objective is not defined and attribution training is not performed.

The multiplier is held fixed within each epoch. At its boundary, it is updated once using an importance-weighted estimate of the training stochastic-policy workload over the full training partition, with independent evaluation draws. Deterministic training and validation workload ratios are monitored separately. An operating point is accepted only if the deployed deterministic policy satisfies its specified budget and quality criteria. Validation violation triggers rejection, not a claim that the stochastic constraint guarantees deterministic feasibility. The requested budget, realized deterministic workload, and stochastic-policy workload are distinct quantities. A cost value alone is not a certificate of constraint satisfaction.

3.3 Predictive Backbone

3.3.1 Encoder and prediction head

Given a user set $\mathcal{U}$, the advertising interaction journey of user i is represented as $J_i = (p_{i1}, p_{i2}, \ldots, p_{iT_i})$. Each touchpoint $p_{it} = (c_{it}, f_{it}^{\text{cat}}, f_{it}^{\text{num}}, m_{it})$ contains a channel identifier, categorical fields, continuous fields, and missingness indicators. Categorical fields use separate embedding tables; continuous fields are transformed using training-partition statistics only and projected through a small MLP:

$$e_{c,it} = \text{Embed}_c(c_{it}), \tag{24}$$

$$e_{f,it}^{\text{cat}} = \underset{r=1}{\overset{R_f}{\text{Concat}}} \text{Embed}_r(f_{it,r}^{\text{cat}}), \tag{25}$$

$$e_{f,it}^{\text{num}} = \text{MLP}_f([\text{Norm}_{\text{train}}(f_{it}^{\text{num}}); m_{it}]), \tag{26}$$

$$e_{f,it} = [e_{f,it}^{\text{cat}}; e_{f,it}^{\text{num}}]. \tag{27}$$

Unknown categories use training-defined out-of-vocabulary tokens, and missing continuous values

are imputed with training statistics while retaining m_{it}. These are concatenated to form the sequential input representation:

$$e_{in,it} = [e_{c,it}; e_{f,it}], \quad e_{in,i} = (e_{in,i1}, \ldots, e_{in,iT_i}) \tag{28}$$

Let $d_{ef} = \dim(e_{f,it})$ denote the encoded feature width, distinct from the raw feature width d_f, and let $d_{in} = \dim(e_{c,it}) + d_{ef}$. The shared LSTM accepts exactly d_{in} features per event and has hidden width d_h. The full path supplies $e_{in,it} \in \mathbb{R}^{d_{in}}$ directly. The selection path supplies the fine-stage output at the same width, as specified below. The full-path recurrence is:

$$h_{it} = \mathrm{LSTM}(e_{in,it}, h_{i,t-1}) \tag{29}$$

Touchpoint-level representations are aggregated via additive attention. For attention width d_a, let $W_h \in \mathbb{R}^{d_a \times d_h}$, $W_e \in \mathbb{R}^{d_a \times d_{in}}$, $b_a, v_a \in \mathbb{R}^{d_a}$:

$$\begin{aligned} s_{it} &= v_a^\top \tanh(W_h h_{it} + W_e e_{in,it} + b_a), \\ \omega_{it} &= \frac{\exp(s_{it})}{\sum_{k=1}^{T_i} \exp(s_{ik})}. \end{aligned} \tag{30}$$

$$m_i = \sum_{t=1}^{T_i} \omega_{it} h_{it} \tag{31}$$

User/request attributes are split into categorical, continuous, and missingness fields using the same training-only preprocessing rule as Eq. (27). Their concatenated representation is projected to the encoded touchpoint-feature width, with $W_u \in \mathbb{R}^{d_{ef} \times \dim(e_{u,i})}$ and $b_u \in \mathbb{R}^{d_{ef}}$:

$$\begin{aligned} e_{u,i} &= \Big[\operatorname*{Concat}_{r=1}^{R_u} \mathrm{Embed}_r(u_{i,r}^{\mathrm{cat}}); \\ &\qquad \mathrm{Norm}_{\mathrm{train}}(u_i^{\mathrm{num}}); m_i^u \Big], \\ v_{u,i} &= \sigma(W_u e_{u,i} + b_u), \quad v_{u,i} \in \mathbb{R}^{d_{ef}}. \end{aligned} \tag{32}$$

$$\begin{aligned} w_{it} &= \frac{\exp\left(e_{f,it}^\top v_{u,i}\right)}{\sum_{k=1}^{T_i} \exp\left(e_{f,ik}^\top v_{u,i}\right)}, \\ v_{\mathrm{attn},i} &= \sum_{t=1}^{T_i} w_{it} e_{f,it}. \end{aligned} \tag{33}$$

The student parameters θ include copies of the Stage-I teacher's compatible embedding, feature-projection, user-context, LSTM, additive-attention, user-pooling, and prediction-head parameters. These are separate trainable copies; the teacher parameters $\bar{\theta}$ remain frozen. The router, budget head, coarse-query/scoring layers, and fine-attention projections have no teacher counterparts and are initialized independently with the recorded run seed. For each new matrix with fan-in n_{in} and fan-out n_{out}, entries are drawn uniformly from $[-a, a]$, where $a = \sqrt{6/(n_{in} + n_{out})}$; new biases are zero. The coarse-score vector uses the same rule as a matrix with one output. No fine-attention weight is described as teacher-pretrained.

The initial Stage-II warm-up jointly trains the student and new selection modules with the existing LOO-based objective and exploration mixture; it is not an additional pretraining stage. A fixed number E_{warm} of epochs and the exploration schedule are specified before the run. The new layers receive direct gradients when they participate in the realized predictor and likelihood-ratio gradients when they determine discrete choices. Teacher weights and cached targets are not updated during warm-up. The full and selected paths then use the compatible prediction-head architecture:

$$v_i^F = [m_i^F; v_{\mathrm{attn},i}^F; v_{u,i}^F], \qquad z_i^F = \mathrm{MLP}_{\bar{\theta}}(v_i^F), \tag{34}$$

$$v_i^R = [m_i^R; v_{\mathrm{attn},i}^R; v_{u,i}^R], \qquad z_i^R = \mathrm{MLP}_{\theta}(v_i^R). \tag{35}$$

Here m_i^F and $v_{\mathrm{attn},i}^F$ encode the complete observed journey, whereas m_i^R and $v_{\mathrm{attn},i}^R$ are recomputed exclusively from the routed touchpoints. Only the raw static prediction-time user/request fields are shared. The teacher retains its own frozen categorical embeddings, continuous-feature projections, user-context encoder, recurrent encoder, pooling layers, output head, and temperature. The student receives separate parameter copies, including its user-context encoder, and can update them without modifying teacher state. Thus $v_{u,i}^F$ and $v_{u,i}^R$ need not coincide after training; the earlier unsuperscripted user-encoder equation defines their common architecture, not shared trainable storage. Training-only preprocessing statistics and category dictionaries are fixed for both. Teacher caches are tied to the frozen teacher and its preprocessing/calibration version. Thus the selection

path cannot recover excluded touchpoints through a full-journey side representation. Touchpoint-level scores are derived from the routed predictor in Sec. 3.5.

For Stage-II/III nonempty inputs, $p_i^F = \sigma(z_i^F/\tau_F)$ and $p_i^R = \sigma(z_i^R/\tau_F)$: the common attribution temperature is $\tau_{\text{attr}} = \tau_F$. These probabilities are used consistently in prediction, distillation, LOO, and Shapley losses. Stage I trains its teacher at unit temperature before calibration. The main student supervision is binary cross-entropy,

$$\mathcal{L}_{\text{pred}} = -\frac{1}{N}\sum_{i=1}^{N}\big[y_i \log p_i^R + (1-y_i)\log(1-p_i^R)\big]. \tag{36}$$

Prediction distillation aligns the student to the stop-gradient full-observed-journey teacher through

$$\mathcal{L}_{\text{KD}} = \frac{1}{N}\sum_i D_{\text{KL}}\big(\text{Bern}(p_i^F)\|\,\text{Bern}(p_i^R)\big). \tag{37}$$

During consistency training, the shared fixed temperature prevents different temperature choices from being conflated with path changes; it does not eliminate differences in learned logit scale. Student prediction and coalition passes in Stage II/III use deterministic layers with dropout disabled; dropout is a Stage-I training option. Routing and subset sampling remain stochastic. After the routed model and budget are fixed, a separate scalar temperature τ_R is fitted to routed validation logits for each operating point. All reported routed predictions and nonempty-coalition values use that frozen τ_R; calibration metrics and final credits are computed from these values. Full-teacher nonempty-coalition outputs retain their own frozen temperature. In both training and reporting, the empty coalition is an explicit exception: teacher and student share the separately calibrated, frozen context-only probability defined in Sec. 3.5, without applying τ_{attr} or τ_R to it again. Temperature scaling follows the validation-based calibration approach of Guo et al. (Guo et al. 2017a).

3.4 Order-Preserving Long-Journey Selection

When the learned router selects the selection expert, LSCM implements index-preserving coarse selection and fine refinement under the journey-specific budget K_i. The implementation is as follows:

Let $\mathbf{X}_i \in \mathbb{R}^{T_i\times(H+C)}$ denote the long-journey feature matrix. We partition it into intrinsic features $\mathbf{X}_i^h \in \mathbb{R}^{T_i\times H}$ (e.g., channel and product descriptors) and interaction cross-features $\mathbf{X}_i^c \in \mathbb{R}^{T_i\times C}$ (e.g., time gap and interaction duration):

$$\mathbf{X}_i = \big[\mathbf{X}_i^h \;;\; \mathbf{X}_i^c\big]. \tag{38}$$

This feature matrix partitions the encoded event representation into disjoint intrinsic and cross-feature columns, so $H + C = d_{in}$; the field-to-column mapping is a dataset-specific input to the method and must be fixed before fitting. The notation separates journey length T_i, intrinsic dimension H, cross-feature dimension C, and routed retained count K_i.

For coarse selection, the query is built only from context available at the prediction timestamp:

$$\mathbf{q}_i = \text{MLP}_q([u_i; a_i^{\text{pre}}]) \in \mathbb{R}^{1\times H}, \tag{39}$$

where a_i^{pre} contains campaign/request context fixed before the outcome window. It excludes the conversion indicator, post-conversion attributes, and any feature derived from the final outcome. For non-converted journeys the same fields and timestamp rule apply. With $\mathbf{W}^h, \mathbf{W}^q \in \mathbb{R}^{H\times d_k}$, $\mathbf{W}^c \in \mathbb{R}^{C\times C}$, and $\mathbf{w}_c \in \mathbb{R}^C$, the relevance vector is

$$\boldsymbol{\zeta}_i = \frac{(\mathbf{X}_i^h\mathbf{W}^h)(\mathbf{q}_i\mathbf{W}^q)^\top}{\sqrt{d_k}} + (\mathbf{X}_i^c\mathbf{W}^c)\mathbf{w}_c \in \mathbb{R}^{T_i}. \tag{40}$$

At deterministic evaluation, the top $K_i < T_i$ indices of $\boldsymbol{\zeta}_i$ are selected with the budget from Eq. (7); score ties use original chronological index. Route ties prefer full processing, and budget ties prefer the smaller feasible budget. During training, Eq. (9) samples distinct indices and Eq. (10) trains the coarse scorer; validation and test inference use deterministic hard Top-K_i. The positive selection temperature is fixed throughout each run; only

the route/budget exploration coefficient follows the prespecified decay schedule. Selected interactions are restored to chronological order before refinement, so retrieval does not replace temporal encoding.

Let $I_i = (j_1 < \cdots < j_{K_i})$ be the selected original indices, where K_i is the feasible hard budget chosen by Eq. (7), and define $\mathbf{X}'_i = \mathcal{F}_\theta(J_i|_{I_i}) \in \mathbb{R}^{K_i \times (H+C)}$, where $\mathcal{F}_\theta$ applies the fixed raw-field reconstruction and training-fitted preprocessing, followed by the student feature embeddings/projections, to the retained chronological events. The preprocessing rule is fixed, while its student embedding/projection parameters remain trainable; intrinsic and cross-feature widths are H and C in this encoded representation. Event-local fields and absolute timestamps are preserved, but event-relative fields, including previous-touch gaps, are recomputed within I_i using the same start-of-journey convention and training-fitted transformations. Thus $\mathbf{X}'_i$ is not a simple slice of precomputed full-journey relative features. The same rebuilt retained events supply user-aware pooling. Coarse scores and routing summaries are computed once on the full observed journey; they are not recomputed after selection. Training, ordinary prediction, LOO, and Shapley evaluation all use this feature-construction rule. The fine stage must preserve one output per retained touchpoint. For head $a \in \{1, \ldots, n_h\}$,

$$\mathbf{Q}_{ia} = \mathbf{X}'_i \mathbf{W}_a^Q, \quad \mathbf{K}_{ia} = \mathbf{X}'_i \mathbf{W}_a^K, \quad \mathbf{V}_{ia} = \mathbf{X}'_i \mathbf{W}_a^V, \tag{41}$$

$$\mathbf{Z}_{ia} = \mathrm{softmax}\left(\frac{\mathbf{Q}_{ia}\mathbf{K}_{ia}^\top}{\sqrt{d_k}} + \mathcal{M}_i^{\mathrm{causal}} \right) \mathbf{V}_{ia}, \tag{42}$$

$$\mathbf{Z}'_i = \mathrm{Concat}(\mathbf{Z}_{i1}, \ldots, \mathbf{Z}_{in_h}) \mathbf{W}^O \in \mathbb{R}^{K_i \times d_{in}}. \tag{43}$$

The causal mask has $\mathcal{M}_{ab}^{\mathrm{causal}} = 0$ for $b \le a$ and $-\infty$ otherwise, using the current retained sequence or coalition size. Query/key projection matrices have shape $(H + C) \times d_k$, and value projections have shape $(H + C) \times d_v$. Each attention head has value width d_v, and $\mathbf{W}^O \in \mathbb{R}^{n_h d_v \times d_{in}}$ projects the concatenated heads to the shared input width. Thus $\mathbf{Z}'_i$ replaces $e_{in,i}$ at the LSTM input and at the input-dependent additive-attention term; no additional adapter is implicit. Hidden width d_h is unchanged.

User-aware pooling uses retained event features only. The retained-feature rebuild, pooling, and input/output projections are included in $c_q K_i$. The index map I_i is retained through prediction and attribution so that every computed score maps to exactly one logged touchpoint. This makes the coarse stage auditable and avoids conflating sequence compression with attention pooling.

Selection audit. Table 6 examines retention, top-credit recall, channel errors, and channel-rank agreement under forced budgets for a fixed student.

3.5 Shapley Credit with Confidence Stopping

Efficient Shapley estimation can use regression-based estimators with uncertainty assessment (Covert and Lee 2021), structured permutation sampling (Mitchell et al. 2022), or an amortized explainer such as FastSHAP (Jethani et al. 2022). Our estimator retains explicit coalition evaluations and controls their sampling precision separately from the routing budget.

To quantify the contribution of individual advertising touchpoints, AdaSeq-MTA first estimates Shapley values for a specified predictive value function. The unrectified exact values satisfy the standard cooperative-game axioms for that game (Shapley 1952). The final non-negative accounting credits in Eq. (2) additionally apply selection, rectification, and renormalization and therefore must not be described as satisfying all Shapley axioms.

Formally, define two distinct predictive games. The full game has player set $A_i^F = \{1, \ldots, T_i\}$ and value $v_i^F(S)$ from the frozen full-observed-journey predictor. The routed game has player set $A_i^R = I_i$ and value $v_i^R(S)$ from the routed predictor. For either $e \in \{F, R\}$ and $t \in A_i^e$, the exact Shapley value is

$$SV_{i,t}^e = \sum_{S \subseteq A_i^e \setminus \{t\}} \frac{|S|!(|A_i^e| - |S| - 1)!}{|A_i^e|!} \times \left[v_i^e(S \cup \{t\}) - v_i^e(S) \right]. \tag{44}$$

Thus routed credit is a Shapley value of its own retained-player game, not an approximation to the

Algorithm 1 Learning procedure of AdaSeq-MTA

Input: Training dataset $\mathcal{D}$; separate validation and held-out evaluation sets; retained budgets $\mathcal{K}$; routing target B; precision $(\epsilon_{\text{touch}}, \epsilon_{\text{channel}}, \delta)$; $M_{\max}$.
Output: Signed contributions, normalized credits, and accounting allocations for mature outcomes.
{Stage I: full-observed-journey teacher}
Train and calibrate the full teacher and context-only baseline; freeze both.
Train and freeze the training-only uncertainty predictor.
Cache teacher predictions and LOO credit proxies.
Compute $B_{\min}$ with Eq. (20); reject an infeasible requested budget.
Copy compatible teacher modules into the student; initialize new router/selection layers independently.
{Stage II: router/student warm-up and constrained training}
for each sampled journey $i \sim q(i)$ in the current training epoch **do**
    Compute $\mathbf{g}_i$ with Eq. (4); mask infeasible experts.
    Sample e_i and feasible K_i from the scheduled mixed distributions.
    if $e_i =$ select **then**
        Select K_i original indices, restore chronology, and retain index map I_i.
    else
        Use the complete observed journey and identity index map.
    end if
    Compute full-teacher and routed-student predictions and credit proxies.
    Accumulate weighted direct and likelihood-ratio gradients; step after each complete minibatch.
end for
Repeat training epochs; update λ_B once per epoch from the stochastic training workload.
Apply the feasible-checkpoint rule; reject a failed operating point, otherwise restore the selected Stage-II checkpoint.
{Stage III: Shapley-aligned refinement}
Select the prespecified stratified training subset and cache teacher targets/permutation IDs.
Skip refinement if the converted subset is empty; otherwise perform the following Stage-III updates.
Sample full-data prediction/cost and subset-credit streams with Eq. (15).
Form the full sampled credit vector, then recompute blocks with Eq. (16).
Add weighted direct and likelihood-ratio gradients; step only after both streams finish.
Fine-tune at lower learning rate; retain epochwise full-data dual updates and deterministic validation checks.
Restore the selected feasible checkpoint (Stage II if refinement was skipped); freeze it and fit τ_R on validation logits.
{Model-based credit calculation}
for each journey (u_i, J_i) in the held-out evaluation set $\mathcal{D}_{\text{eval}}$ **do**
    Apply deterministic routing and K_i with the calibrated student.
    Use exact singleton credit; otherwise sample until both confidence tolerances are met or $M_{\max}$.
    Flag journeys reaching the sample cap without both conditions.
    Retain the exact or sampled signed contribution for each retained touchpoint;
    Map signed values back and normalize; form accounting allocations only after y_i matures, using Eq. (2).
end for
return Signed values and normalized credits on $\mathcal{D}_{\text{eval}}$; accounting allocations for mature outcomes.

full-teacher game merely because excluded positions are assigned structural zeros. The selected refinement and fitted parameters change the value function as well as the player set. Full-versus-routed distortion is evaluated after mapping both games to the original T_i positions.

Coalition evaluation contract. The route e_i, retained budget K_i, and original index set I_i are chosen once from the complete observed journey and then held fixed for all coalition evaluations. For $S \subseteq I_i$, the selected-path value is obtained by rebuilding event-relative features from the raw chronological events in S, recomputing the fine-attention projections and causal attention on S, and rerunning the student sequence encoder and prediction head. The router and coarse Top-K selector are not rerun. Full-path teacher and full-path student games likewise recompute their encoders from the raw coalition. No contextualized representation computed on a larger coalition is reused. Event-local embeddings may be cached only when they do not depend on the coalition. Dropout and other inference randomness are disabled. The full retained coalition uses exactly the same feature construction, pooling, deterministic layers, and calibration as ordinary routed prediction, so $v_i^R(I_i) = p_i^R$ at the corresponding training or reporting temperature. Consequently these values explain the realized, fixed processing path, rather than attributing changes in the routing decision itself.

Random permutations are used only to sample predecessor sets; they never reorder the sequence presented to the encoder. Coalition members are restored to their logged chronological order. Absolute timestamps, calendar fields, and event-local attributes remain fixed. Features defined relative to another event, such as previous-touch gap, are not carried over unchanged: the primary value function recomputes them from the immediately preceding retained event using the

same training-only transformation, with a start-of-journey value fixed in the preprocessing configuration and shared across ordinary and coalition evaluation.

For $S = \varnothing$, the primary value is a separately trained context-only predictor using the same static prediction-time user/request fields and no journey token. Let its logit be z_i^0 and its separately fitted validation temperature be τ_0. The common baseline is $v_i^F(\varnothing) = v_i^R(\varnothing) = \sigma(z_i^0/\tau_0)$. Both the context-only predictor and τ_0 are frozen with the teacher; no teacher or student temperature is subsequently applied to this probability. The same baseline is used for the student's forced-full game, LOO targets, and Shapley targets in both training and reporting. Nonempty coalitions follow their respective calibration rules above.

Precision-controlled order-preserving approximation. Exact evaluation of Eq. (44) is exponential. We draw permutations in batches and use predecessor sets only as coalition masks (Singal et al. 2019); every coalition is restored to logged chronological order. For the realized sample count M_i of the game being evaluated (written M_i^e when distinguishing games),

$$\begin{aligned}\widehat{SV}_{i,t}^{e}(M_i) &= \frac{1}{M_i}\sum_{m=1}^{M_i}\Delta_{it}^{e,(m)},\\ \Delta_{it}^{e,(m)} &= v_i^e\left(S_{\pi^{(m)},t}\cup\{t\}\right)\\ &\quad - v_i^e\left(S_{\pi^{(m)},t}\right).\end{aligned} \tag{45}$$

For either game e, let $L_i^e = |A_i^e|$, let C_i^e count its represented channels, and let n_{ic}^e count its players in channel c. Thus $L_i^F = T_i$ and $L_i^R = K_i^{\text{eff}}$. Permutations are sampled independently and uniformly with replacement. Require $M_{\max} \geq 2$, positive tolerances, and $0 < \delta < 1$. Check at the distinct increasing values $M_j = \min(2^j, M_{\max})$, starting at $j = 1$ and ending at the first occurrence of the cap. Allocate $\delta_{ij}^e = 6\delta/[\pi^2 j^2 (L_i^e + C_i^e)]$. The empirical-Bernstein half-width for each player is (Maurer and Pontil 2009)

$$h_{it}^e(M_j) = \sqrt{\frac{2\widehat{V}_{it}^e(M_j)\log(4/\delta_{ij}^e)}{M_j}} + \frac{14\log(4/\delta_{ij}^e)}{3(M_j-1)}, \tag{46}$$

where $\widehat{V}_{it}^e$ is the unbiased sample variance with denominator $M_j - 1$ of marginals in $[-1, 1]$. For channel c, aggregate within each permutation before computing variance:

$$\begin{aligned}G_{ic}^{e,(m)} &= \sum_{t\in A_i^e: c_{it}=c}\Delta_{it}^{e,(m)},\\ h_{ic}^e(M_j) &= \sqrt{\frac{2\widehat{V}_{ic}^e(M_j)\log(4/\delta_{ij}^e)}{M_j}}\\ &\quad + \frac{14 n_{ic}^e\log(4/\delta_{ij}^e)}{3(M_j-1)}.\end{aligned} \tag{47}$$

Here $G_{ic}^{e,(m)} \in [-n_{ic}^e, n_{ic}^e]$, and its unbiased sample variance retains within-permutation covariance. Its expectation is the sum of touchpoint Shapley values in channel c, not generally the Shapley value of a different game that merges that channel into one player. Stop when all player half-widths are at most ϵ_{touch} and all represented-channel half-widths are at most $\epsilon_{\text{channel}}$, or at the cap. The two-sided bound follows by rescaling the $[0, 1]$ empirical Bernstein inequality and allocating $\delta_{ij}^e/2$ to each tail. Conditional on the fitted model, calibration, journey, and fixed route, a union bound gives

$$\sum_{j\geq 1}(L_i^e + C_i^e)\delta_{ij}^e = \frac{6\delta}{\pi^2}\sum_{j\geq 1} j^{-2} = \delta. \tag{48}$$

Consequently all these signed intervals cover simultaneously with probability at least $1 - \delta$, including at the selected checkpoint. This is a per-journey, per-game guarantee; joint claims for several games or journeys need an additional error allocation. It does not cover model misspecification, out-of-support coalitions, rectification, normalization, or causal error. Cap-hit cases without both tolerance conditions are flagged.

Precision feasibility before sampling. For each nonsingleton game, compute $a_j = 14\log(4/\delta_{ij}^e)/[3(M_j-1)]$ at every permitted checkpoint and let $a_{\min} = \min_j a_j$. The touchpoint half-width cannot be smaller than a_j, and channel c has lower bound $n_{ic}^e a_j$, even at zero sample variance. If $a_{\min} > \epsilon_{\text{touch}}$ or $(\max_c n_{ic}^e)a_{\min} > \epsilon_{\text{channel}}$, no checkpoint can certify both tolerances under this bound. Such configurations cannot certify the requested precision under the stated cap and tolerances. Satisfying this necessary condition does not guarantee stopping, because empirical variance also contributes. Any revised cap or tolerance

defines a separately declared configuration and is never chosen from test results. Equal sample counts alone do not imply matched precision.

The naive evaluation count is $M_i^e(L_i^e + 1)$. With the common empty value and each game's full value computed once, at most $2 + M_i^e(L_i^e - 1)$ evaluations are needed before any additional exact-coalition cache reuse. For a singleton, its contribution is exactly $v_i^e(A_i^e) - v_i^e(\varnothing)$ and needs no permutation sampling. Exact singleton evaluation is distinct from Monte Carlo stopping. Raw-call counts, cache hits, sample counts, and latency describe different aspects of estimator cost. Under selection, scores map back through I_i, and excluded positions remain zero. Common random permutations across models are obtained by restricting full-journey permutations to each player set, preserving each game's uniform marginal sampling law.

We set $\hat{r}_{it} = \widehat{SV}^R_{i,t}$ for retained touchpoints and $\hat{r}_{it} = 0$ with an exclusion indicator for filtered touchpoints. Signed values are retained as diagnostic contributions so suppressive or saturation effects are not erased. Negative retained scores are rectified only when constructing the separate non-negative accounting allocation in Eq. (2); no sigmoid normalization is used. If every retained score is non-positive on a converted journey, the accounting fallback is uniform over retained original touchpoints, never over LSCM-excluded positions, while the signed diagnostic remains unchanged. The complete workflow is detailed in Algorithm 1.

3.6 Marketing Decision-Support Workflow

AdaSeq-MTA can support the following staged decision workflow. First, historical impressions are assembled into leakage-safe customer journeys and processed by the learned evidence-aware policy under an explicit computational budget. Second, model-based touchpoint credits are aggregated into channel-level summaries using only the training partition. Third, the summaries generate a candidate budget allocation. Fourth, the candidate is screened on held-out logged support.

For non-overlapping accounting journeys with each observed conversion counted once, a descriptive aggregate credit for channel c is

$$R_c = \frac{\sum_i \sum_{t:c_{it}=c} \alpha_{it}}{\sum_i \sum_{t:c_{it}=c} \mathrm{cost}_{it}}, \tag{49}$$

when logged costs are available and positive.

4 Experiments

The evaluation covers prediction, credit agreement, and computational performance. The experiments address the following questions through prediction, attribution, ablation, and efficiency comparisons.

1. What prediction, credit-drift, and latency trade-offs do the evaluated routing policies exhibit?

2. How do the full routing configuration and its objective variants compare with restricted routing configurations and refinement controls?

3. How do the evaluated methods compare in touchpoint/channel credit error, ranking agreement, and seed stability within the attribution comparison?

4. How do adaptive modules affect prediction, latency, parameter count, and memory on the tested LSTM and Transformer backbones?

5. What sampling-count, credit-drift, and attribution-latency trade-offs are observed for fixed-sample and adaptive-stopping settings?

Masked-view, adversarial, replacement, and coalition-support studies are outside these core research questions.

4.1 Experiment Settings

The experiments use the public Criteo Attribution Modeling for Bidding dataset and a secondary synthetic prediction benchmark. The Criteo release contains approximately 16.5 million impressions across 700 campaigns[1], with nine anonymized categorical covariates and user, campaign, time, feedback, and cost fields (Diemert et al. 2017; Kumar et al. 2020). Campaign IDs define the channel-credit grouping. The

[1] Criteo AI Lab: dataset description and field definitions.

prediction-time input contract excludes outcome-derived fields, including conversion flags, conversion timestamps and IDs, advertiser-attribution flags, conversion-relative click positions/counts, and cost-per-order. Other feedback is admissible only when observed by the prediction timestamp. Relative event features are rebuilt after selection. Dataset statistics appear in Table 8.

The primary evaluation uses a forward-time split. Preprocessing is fitted on training data, and model selection and calibration use validation data only. For prediction time s_i, inputs lie in $[s_i - W, s_i]$, and the outcome window is $(s_i, s_i + H_y]$. With label delay d_y, training labels satisfy $s_i + H_y + d_y \leq c_{\text{tr}}$, and validation labels used for selection are available by test start c_{te}. Incomplete outcome windows are excluded rather than labeled negative. The release's supplied conversion flag uses a 30-day horizon. Synthetic prediction, teacher agreement, and external-credit evaluation have distinct targets. External-credit evaluation requires targets independent of the fitted models and aligned to original events; MAC's mechanism-generated conversion labels (Wu et al. 2026) alone do not define touchpoint-credit ground truth.

Each training stage uses Adam (Kingma and Ba 2015) for at most 50 epochs, with five-epoch early-stopping patience; Stage II activates patience after its fixed warm-up. Validation search covers learning rates $\{10^{-4}, 10^{-3}, 10^{-2}\}$, batch sizes $\{128, 256, 512\}$, hidden widths $\{64, 128, 256\}$, and dropout $\{0.1, 0.2, 0.3\}$. Retained budgets are $\mathcal{K} = \{1, 2, 4, 8, 16\}$, masked by $K < T_i$. Workload budgets, exploration schedules, loss coefficients, and precision tolerances are validation-selected. Checkpoints minimize validation log-loss; student checkpoints additionally satisfy the workload budget, with ties resolved by lower workload and then earlier epoch. Calibration follows checkpoint selection. In Stages II/III, an epoch comprises $\lceil N/m_D \rceil$ updates with replacement; Stage III additionally draws m_S refinement samples per update. Stage III restores the selected Stage-II parameters and dual state, resets Adam moments, and uses a lower learning rate. The LOO-continuation control shares this initialization and schedule.

Learned-model results summarize five fixed training seeds as mean ± standard deviation, except for the full-policy overall log-loss, which is reported as a length-weighted point estimate; deterministic rule references and pairwise Seed JS summaries report point estimates only. Prediction metrics include ROC-AUC, PR-AUC, log-loss and ECE. Attribution metrics include touchpoint/channel MAE, rank correlation, top-k agreement, and cross-seed stability; efficiency metrics include latency, throughput, memory, and sampling cost. Seed standard deviations and displayed intervals are descriptive uncertainty summaries; no paired non-inferiority conclusion is drawn. Undefined metrics are N/A rather than zero. The detailed credit metrics and timing conventions are described in Sec. 4.2.

4.2 Evaluation Protocol and Reporting

Evaluation procedure. All learned models use a common Shapley evaluation procedure, map credits to the original touchpoints, and convert them into non-negative, normalized allocations. Each model uses its validation-fitted calibration temperature; paired controls within each seed share the frozen teacher and context-only baseline. Last-touch and linear attribution serve as rule-based references and are reported as point estimates.

Attribution metrics. Touchpoint MAE is averaged first over the original touchpoints within each journey and then over journeys; channel MAE uses the same observed-channel set across methods. Spearman correlation measures agreement in touchpoint-credit rankings, with average ranks for ties and N/A for singleton journeys or constant vectors. Top-k agreement is the overlap fraction between the predicted and target sets of the top $k_i = \min(k, T_i)$ touchpoints, with ties resolved by original chronological order; it equals one when $T_i \leq k$. Cross-seed stability is measured by Seed JS: channel-credit JS divergence is averaged over the ten pairs formed by five training seeds within each identical converted journey, then over journeys. JS uses natural logarithms without distribution smoothing; lower values indicate greater stability. The full pipeline is retrained for each seed. Evaluation uses common permutation streams and a fixed sample count, with exact computation for singleton journeys.

Sampling and timing. With the predictor, calibration, route, budget, and retained indices

fixed, sampling MAE measures absolute differences between signed contributions and a same-game Shapley reference, averaged first over players and then over journeys. Exact enumeration has no Monte Carlo error, while an independent finite-permutation reference retains sampling uncertainty. Sampling MAE, credit drift, and external attribution error are reported separately. Attribution timing excludes reference construction. Each invocation starts with a fresh coalition cache and reuses only results from that invocation; all configurations use the same endpoint and event-local cache rules. Device warm-up is separate, and accelerators are synchronized at timing boundaries. Adaptive stopping reports mean and 95th-percentile (P95) sample counts.

The following tables summarize the reported experimental results, covering routing-policy comparisons, objective and component ablations, length-stratified performance, attribution quality, and computational efficiency. Numerical values and their accompanying uncertainty summaries are presented at the reported precision.

Table 1 shows that AdaSeq-MTA reduces latency from 18.4 to 13.1 ms relative to uniform full-sequence processing, a 28.8% reduction, while retaining 66% of touchpoints. ROC-AUC changes from 0.7758 to 0.7753 and log-loss from 0.1098 to 0.1149. This operating point illustrates a prediction–cost trade-off rather than an improvement in every predictive metric. Relative to prediction-only adaptive routing, AdaSeq-MTA has lower reported channel-credit drift (0.019 versus 0.029) and latency (13.1 versus 13.5 ms). These are comparisons of the displayed operating points, not a statistical non-inferiority conclusion.

Figure 4 shows prediction performance, channel-credit drift, and inference latency across routing strategies. Compared with the prediction-only adaptive policy, AdaSeq-MTA increases AUC from 0.7748 to 0.7753, reduces channel-credit drift from 0.029 to 0.019, and lowers inference latency from 13.5 to 13.1 ms.

To compare the two credit targets, the LOO-continuation control and the full method continue training from the same Stage-II model, with identical training samples, sampling procedures, learning-rate schedules, and update counts, using LOO and Shapley marginals, respectively.

Router stability measures agreement across models trained with different seeds in the processing path and retained-touchpoint count for the same journey. Touchpoint retention is the proportion of original touchpoints retained across all evaluation journeys.

SSTAM, GRL, channel replacement, and coalition training are outside the main experimental comparison.

Adding the adaptive modules increases mean AUC by 0.0051 for LSTM and 0.0023 for Transformer, indicating improved predictive performance on both backbones.

Figure 5 shows the corresponding changes in computational resources: inference latency decreases on both backbones, while parameter count and peak memory increase. These results show that adaptive processing achieves higher predictive AUC and lower inference latency with increased model capacity and memory use.

Table 5 presents attribution quality and stability across training seeds. Compared with prediction-only adaptive routing, AdaSeq-MTA reduces touchpoint MAE from 0.1640 to 0.1510, a reduction of 7.9%, and increases Spearman correlation from 0.5730 to 0.6060. Touchpoint MAE is also 3.2% lower than that of the AD-KD-inspired control. Its channel-credit MAE is 0.1180, below 0.1300 and 0.1220 for the two controls, indicating that the credit-consistency constraint helps improve credit allocation at both touchpoint and channel levels.

AdaSeq-MTA has Seed JS of 0.0260, below 0.0330 for prediction-only adaptive routing and 0.0280 for the AD-KD-inspired control, indicating more stable channel-credit allocations across training seeds. Together with the error and ranking metrics, these results demonstrate the method's advantages in attribution quality and result stability.

Table 6 compares the effects of touchpoint retention on prediction and attribution. Within each training seed, the model and calibration remain fixed while the retention budget K varies; the full sequence is retained when $K \geq T_i$. Prediction metrics use all evaluation journeys, while attribution metrics use the converted subset and the same model's full-sequence results as the reference. Error bars show standard deviations across training seeds.

Table 1 Routing-policy comparison in prediction, channel-credit drift, touchpoint retention, attribution evaluations, and latency. A smaller backbone retains the full sequence because reducing network width does not remove touchpoints.

Policy	ROC-AUC	PR-AUC	Log-loss	Channel-credit drift	Retained (%)	Attribution evals	Latency (ms)
Uniform full sequence	0.7758 ±0.0017	0.4310 ±0.0029	0.1098 ±0.0008	0.000 [0.000, 0.000]	100	201	18.4
Fixed length thresholds	0.7702 ±0.0010	0.4215 ±0.0021	0.1136 ±0.0011	0.041 [0.038, 0.044]	62	129	12.7
Learned router: length only	0.7715 ±0.0021	0.4240 ±0.0032	0.1128 ±0.0005	0.038 [0.035, 0.041]	64	132	12.9
Learned router: uncertainty only	0.7731 ±0.0014	0.4272 ±0.0026	0.1115 ±0.0013	0.031 [0.028, 0.034]	71	145	13.8
Random feasible routing	0.7624 ±0.0009	0.4090 ±0.0019	0.1182 ±0.0007	0.067 [0.063, 0.071]	58	118	11.9
Teacher-guided fixed-K	0.7740 ±0.0019	0.4288 ±0.0030	0.1109 ±0.0009	0.022 [0.020, 0.024]	67	137	13.2
Prediction-only adaptive policy	0.7748 ±0.0012	0.4296 ±0.0023	0.1103 ±0.0010	0.029 [0.026, 0.032]	69	141	13.5
Uniform smaller backbone	0.7671 ±0.0022	0.4162 ±0.0027	0.1164 ±0.0004	0.046 [0.042, 0.050]	100	113	10.8
AD-KD-inspired adaptive control	0.7741 ±0.0016	0.4282 ±0.0020	0.1107 ±0.0012	0.024 [0.022, 0.026]	68	135	13.3
AdaSeq-MTA full policy	0.7753 ±0.0013	0.4302 ±0.0024	0.1149	0.019 [0.017, 0.021]	66	136	13.1

Table 2 Objective ablations: prediction, touchpoint and channel credit errors, cost, and router stability.

Variant	ROC-AUC	Touch-credit MAE	Channel-credit MAE	Cost	Router stability
No learned router	0.7702 ± 0.0015	0.142 ± 0.0042	0.121 ± 0.0032	1.00	–
Fixed K	0.7715 ± 0.0009	0.138 ± 0.0031	0.118 ± 0.0041	0.72	0.81 ± 0.0212
Without prediction distillation	0.7671 ± 0.0017	0.146 ± 0.0058	0.125 ± 0.0018	0.70	0.79 ± 0.0178
LOO targets only (no Stage III)	0.7738 ± 0.0011	0.134 ± 0.0027	0.115 ± 0.0047	0.71	0.83 ± 0.0234
Without mapped-credit consistency	0.7729 ± 0.0013	0.151 ± 0.0063	0.129 ± 0.0025	0.69	0.82 ± 0.0191
Distribution-only consistency	0.7736 ± 0.0016	0.140 ± 0.0047	0.120 ± 0.0038	0.70	0.83 ± 0.0247
Without signed-marginal term	0.7744 ± 0.0010	0.137 ± 0.0022	0.117 ± 0.0013	0.70	0.84 ± 0.0163
Without positive-mass term	0.7745 ± 0.0018	0.136 ± 0.0054	0.116 ± 0.0044	0.70	0.84 ± 0.0225
Fixed cost penalty (no dual update)	0.7722 ± 0.0012	0.139 ± 0.0036	0.119 ± 0.0029	0.73	0.80 ± 0.0182
LOO continuation (same update budget)	0.7741 ± 0.0014	0.132 ± 0.0069	0.113 ± 0.0035	0.71	0.84 ± 0.0206
AdaSeq-MTA full objective	0.7753 ± 0.0013	0.126 ± 0.0029	0.107 ± 0.0021	0.71	0.86 ± 0.0154

Figure 6 shows that increasing K from 1 to 16 raises top-credit recall from 0.41 to 0.93 and channel-rank agreement from 0.52 to 0.92, while reducing channel-credit MAE from 0.132 to 0.027. Retaining more touchpoints improves agreement with the full-sequence attribution reference.

With a fixed budget of 100 Shapley samples, LSCM reduces attribution latency from 156.4 ms for full-sequence processing to 104.2 ms, a reduction of 33.4%. Adaptive stopping uses a mean of 126 samples, with attribution latency of 126.9 ms and credit drift of 0.0175. Compared with

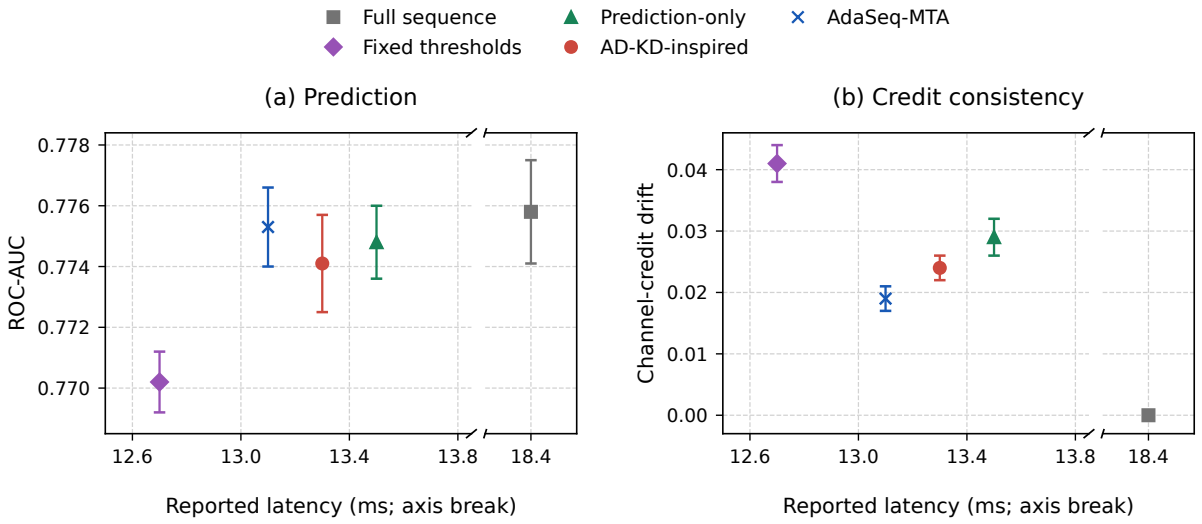


Fig. 4 Trade-offs among prediction performance, attribution consistency, and inference efficiency across routing strategies.

Table 3 Length-stratified performance. Overall log-loss is a descriptive weighted aggregate of the displayed group means, using journey shares 0.412, 0.435, and 0.153; its unrounded value is 0.1149091. This aggregate is also used for the corresponding full-policy entries in the routing, component, and backbone comparisons and is reported as a point estimate.

Group	Journeys (%)	Conversion (%)	ROC-AUC	PR-AUC	Log-loss	ECE
Short	41.2	3.8	0.7624 ± 0.0017	0.3920 ± 0.0028	0.1210 ± 0.0013	0.0310 ± 0.0014
Medium	43.5	4.6	0.7749 ± 0.0012	0.4360 ± 0.0021	0.1115 ± 0.0006	0.0260 ± 0.0008
Long	15.3	5.9	0.7816 ± 0.0019	0.4620 ± 0.0016	0.1082 ± 0.0010	0.0240 ± 0.0011
Overall	100	4.5	0.7753 ± 0.0013	0.4302 ± 0.0024	0.1149	0.0270 ± 0.0006

fixed 100-sample evaluation, it reduces credit drift from 0.0190 to 0.0175. Compared with fixed 200-sample evaluation, it reduces latency from 206.5 ms to 126.9 ms, a reduction of 38.5%, while credit drift increases from 0.0130 to 0.0175. These results show that adaptive stopping provides an operating choice between the two fixed-sample configurations that balances attribution consistency and computational cost.

Table 7 summarizes training time, prediction throughput, peak memory, Shapley sample counts, credit drift, and attribution latency. Adaptive sample counts are reported as the mean and the 95th percentile (P95), reflecting variation in sampling requirements across journeys. Credit drift measures changes in credit allocation relative to the full-sequence reference, while sampling precision is controlled by the confidence bounds in Sec. 3.5.

4.3 Ablation Analysis of Credit-Consistency Objectives

Figure 7 shows the effects of credit-consistency objectives on touchpoint and channel MAEs. Compared with LOO continuation under the same update budget, the full objective reduces touchpoint MAE from 0.132 to 0.126, a 4.5% reduction, and channel MAE from 0.113 to 0.107, a 5.3% reduction.

Removing credit consistency increases touchpoint and channel MAEs to 0.151 and 0.129, respectively. Distribution-only consistency and the variants omitting either the signed-marginal

Table 4 Component configurations and selection controls. Augmentation rows are separate configurations. Target-aware selection and context-score selection are distinct methods.

Variant	Overall ROC-AUC	Short ROC-AUC	Long ROC-AUC	Log-loss	Parameters	Inference ms/journey
Base encoder	0.7702 ±0.0013	0.7580 ±0.0020	0.7760 ±0.0014	0.1136 ±0.0009	1.2M	17.9
+ random/zero padding	0.7640 ±0.0016	0.7520 ±0.0018	0.7700 ±0.0012	0.1178 ±0.0007	1.2M	18.1
+ interpolation augmentation	0.7685 ±0.0012	0.7560 ±0.0022	0.7740 ±0.0015	0.1148 ±0.0010	1.3M	18.3
+ masked augmentation	0.7721 ±0.0017	0.7590 ±0.0017	0.7775 ±0.0011	0.1123 ±0.0008	1.3M	18.5
+ recent-K truncation	0.7690 ±0.0011	0.7565 ±0.0021	0.7745 ±0.0016	0.1145 ±0.0011	1.3M	12.8
+ target-aware top-K	0.7735 ±0.0015	0.7600 ±0.0019	0.7790 ±0.0013	0.1109 ±0.0006	1.4M	13.0
Random feasible fixed-K	0.7662 ± 0.0014	0.7541 ± 0.0016	0.7722 ± 0.0010	0.1160 ± 0.0012	1.3M	12.6
Teacher-guided fixed-K	0.7740 ± 0.0018	0.7608 ± 0.0023	0.7795 ± 0.0017	0.1109 ± 0.0005	1.4M	13.0
Prediction-only adaptive policy	0.7748 ± 0.0010	0.7614 ± 0.0015	0.7802 ± 0.0009	0.1103 ± 0.0013	1.4M	13.1
Context-score fixed-K	0.7727 ± 0.0019	0.7592 ± 0.0024	0.7778 ± 0.0018	0.1118 ± 0.0004	1.3M	12.9
AdaSeq-MTA core	0.7753 ±0.0013	0.7620 ±0.0014	0.7810 ±0.0008	0.1149	1.5M	13.1

term or the positive-mass term also yield higher errors on both measures than the full objective. These results indicate that the credit-consistency terms jointly help reduce attribution error. Error bars show standard deviations across training seeds.

4.4 Computational Cost and Efficiency Analysis

We analyze computational cost at two levels: individual journeys and the evaluation workload. Coarse scoring costs $O(THd_k+TC^2+Hd_k)$ before partial sorting; fine projections, attention products, and recurrent encoding cost $O(Kn_h[(H+C)(2d_k+d_v)+d_vd_{in}])$, $O(K^2n_h(d_k+d_v))$, and $O(Kd_h(d_{in}+d_h))$, respectively. Here H and C are the intrinsic and cross-feature widths, d_k and d_v are the projection and value widths, and n_h is the number of attention heads. With architecture widths fixed, the non-negative coefficients in Eq. (18) summarize these costs. Let $F(T)$ and $P(T,K)$ denote the operation costs of full-sequence and selected-sequence processing. Their ratio gives the analytical speedup:

$$\begin{aligned} F(T) &= c_0 + c_r + (c_f + c_e)T, \\ P(T,K) &= c_0 + c_r + (c_f + c_p)T \\ &\quad + c_sT\log(1+K) \\ &\quad + (c_e + c_q)K + c_aK^2, \\ S_{\text{select}}(T,K) &= F(T)/P(T,K). \end{aligned} \tag{50}$$

Under this proxy, selection saves operations exactly when

$$\begin{aligned} c_e(T-K) > c_pT + c_sT\log(1+K) \\ + c_qK + c_aK^2. \end{aligned} \tag{51}$$

The left side is the recurrent computation saved by retaining fewer touchpoints; the right side is the additional cost of scoring, selection, and fine attention. Selection reduces operations when the savings exceed this overhead. The coefficients

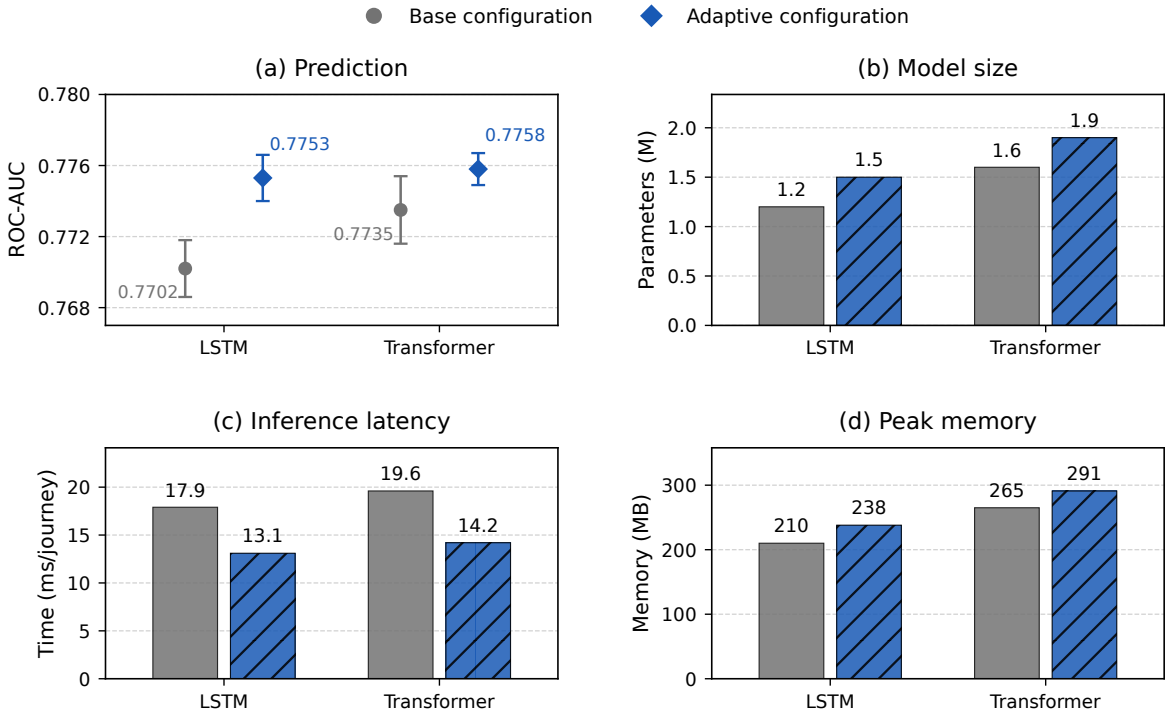


Fig. 5 Prediction and resource use with and without adaptive modules on each backbone. AUC error bars show seed standard deviations; configurations are not parameter-matched.

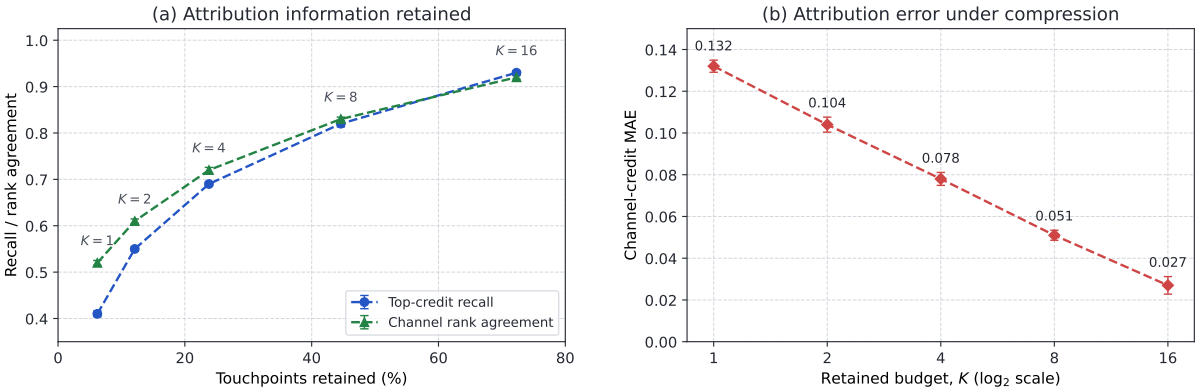


Fig. 6 Fixed-model selection audit from Table 6: recall, channel-rank agreement, and MAE relative to the forced-full reference. Error bars show seed standard deviations.

are obtained from training-only operation counts or hardware calibration, with architecture widths fixed and the fine-output projection included. Contextual states are recomputed for each coalition. Actual runtime also depends on event-local caching, batching, and hardware utilization.

Table 5 Attribution quality and seed stability. Spearman correlation for uniform per-journey linear credit is undefined and shown as N/A.

Model	Touchpoint MAE	Spearman ρ	Channel-credit MAE	Top-k agreement	Seed JS (lower is better)
Last touch	0.2210	0.4120	0.1870	0.3800	–
Linear attribution	0.1980	N/A	0.1620	0.4400	–
CausalMTA	0.1760 ± 0.0023	0.5430 ± 0.0043	0.1410 ± 0.0032	0.5100 ± 0.0041	0.0410
DCRMTA (Tang 2024)	0.1690 ± 0.0027	0.5610 ± 0.0051	0.1360 ± 0.0026	0.5300 ± 0.0045	0.0380
Full-observed-journey teacher	0.1580 ± 0.0021	0.5920 ± 0.0047	0.1240 ± 0.0022	0.5700 ± 0.0039	0.0290
Prediction-only adaptive policy	0.1640 ± 0.0033	0.5730 ± 0.0062	0.1300 ± 0.0034	0.5420 ± 0.0054	0.0330
AD-KD-inspired adaptive control	0.1560 ± 0.0019	0.5850 ± 0.0041	0.1220 ± 0.0020	0.5610 ± 0.0037	0.0280
AdaSeq-MTA	0.1510 ± 0.0025	0.6060 ± 0.0055	0.1180 ± 0.0028	0.5950 ± 0.0050	0.0260

Table 6 LSCM selection audit. Full-path rank agreement applies only to valid nonconstant reference vectors.

Retained K	Touchpoints retained	Top-credit recall	Channel JS	Channel-credit MAE	Rank agreement	Long-group AUC
1	6.2%	0.4100 ± 0.0063	0.1180 ± 0.0034	0.1320 ± 0.0029	0.5200 ± 0.0052	0.7540 ± 0.0018
2	12.1%	0.5500 ± 0.0051	0.0890 ± 0.0027	0.1040 ± 0.0036	0.6100 ± 0.0047	0.7610 ± 0.0015
4	23.8%	0.6900 ± 0.0046	0.0610 ± 0.0022	0.0780 ± 0.0031	0.7200 ± 0.0058	0.7690 ± 0.0012
8	44.6%	0.8200 ± 0.0068	0.0370 ± 0.0039	0.0510 ± 0.0024	0.8300 ± 0.0043	0.7760 ± 0.0019
16	72.3%	0.9300 ± 0.0041	0.0180 ± 0.0025	0.0270 ± 0.0042	0.9200 ± 0.0038	0.7800 ± 0.0014
Full sequence	100%	100%	0	0	1	0.7810 ± 0.0016

Table 9 summarizes journey lengths, retention budgets, compression eligibility, and analytical operation ratios.

For the evaluation workload, let $\mathcal{I}_s = \{i : e_i = \text{select}\}$ be the journeys using selection, with $F_i = F(T_i)$ and $P_i = P(T_i, K_i)$. Their share of full-path operation cost, ρ_s, and aggregate speedup, S_s^{agg}, are

$$\rho_s = \frac{\sum_{i \in \mathcal{I}_s} F_i}{\sum_i F_i}, \qquad S_s^{\text{agg}} = \frac{\sum_{i \in \mathcal{I}_s} F_i}{\sum_{i \in \mathcal{I}_s} P_i}. \tag{52}$$

Both quantities are computed from total operation costs on the same workload. Combining selected and full-sequence journeys gives the overall analytical speedup:

$$\begin{aligned} S_{\text{proxy}} &= \frac{\sum_i F_i}{\sum_{i \notin \mathcal{I}_s} F_i + \sum_{i \in \mathcal{I}_s} P_i} \\ &= \left(1 - \rho_s + \frac{\rho_s}{S_s^{\text{agg}}}\right)^{-1}. \end{aligned} \tag{53}$$

When no journey uses selection, $S_{\text{proxy}} = 1$. The calculation includes feature extraction, routing, selection, and encoding costs, using observed journey lengths and realized retention budgets. It describes operation savings; measured latency additionally reflects data movement, batching, hardware utilization, and separately amortized teacher construction.

For attribution, the naive evaluation count is $M_i(K_i^{\text{eff}} + 1)$. Endpoint reuse and exact singleton evaluation reduce actual calls as described in

Table 7 Efficiency comparison across processing paths and sampling settings. Credit drift and same-game sampling MAE are distinct metrics.

Setting	Train min/epoch	Reported journeys/s	Peak memory	Shapley samples	Reported credit drift	Attribution ms/journey
Full sequence	4.2	182.0	312 MB	25	0.0410 ± 0.0019	42.1
Full sequence	4.2	181.5	318 MB	100	0.0210 ± 0.0013	156.4
LSCM	3.1	246.0	236 MB	25	0.0380 ± 0.0022	28.7
LSCM	3.1	245.5	241 MB	100	0.0190 ± 0.0016	104.2
LSCM	3.1	245.0	247 MB	200	0.0130 ± 0.0011	206.5
Adaptive stopping	3.1	244.0	244 MB	mean/P95 126/188	0.0175 ± 0.0024	126.9

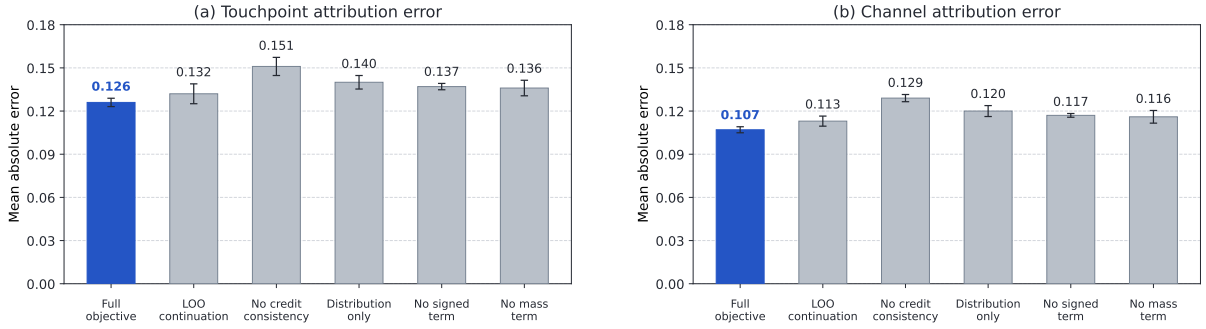


Fig. 7 Objective-variant MAEs from Table 2. Error bars show seed standard deviations; LOO continuation matches the update budget of the full objective.

Table 8 Dataset statistics. The Criteo mean journey length is rounded from 16,500,000 touchpoints divided by 8,600,000 journeys.

Statistics	Criteo	Synthetic
No. users	1,240,000	50,000
No. campaigns	675	20
No. journeys	8,600,000	200,000
No. converted journeys	387,000	18,000
No. touchpoints	16,500,000	1,200,000
Avg. journey length	1.92	6.00
Conversion rate	4.5%	9.0%

Sec. 3.5. The measured attribution latency and adaptive sample counts in Table 7 complement this cost analysis by showing runtime under the evaluated retention budgets and stopping settings.

5 Conclusion

AdaSeq-MTA introduces an adaptive multi-touch attribution framework that jointly considers prediction performance, attribution consistency, and computational efficiency. The framework is trained with a joint prediction, credit-consistency, and computational-cost objective. It selects full processing or touchpoint selection according to journey characteristics and constrains teacher–student attribution differences at the original touchpoint positions. Confidence-controlled Shapley sampling further provides an adaptive balance between estimation precision and attribution computation cost.

Experimental results show that, compared with uniform full-sequence processing, AdaSeq-MTA reduces inference latency from 18.4 ms to 13.1 ms, a reduction of 28.8%, while retaining comparable prediction performance, with ROC-AUC values of 0.7758 and 0.7753, respectively. Compared with the prediction-only adaptive policy, its channel-credit drift decreases from 0.029

Table 9 Journey lengths, retention budgets, and analytical operation ratios.

T source	Reported T	Reported K	Journeys with $K < T_i$	Operation ratio
Median	2	1	0.41	0.72
P90	5	4	0.68	0.81
P95	8	4	0.79	0.76
P99	17	8	0.91	0.69
Maximum	64	16	0.98	0.58

to 0.019, illustrating the role of the credit-consistency constraint. Results on two backbone networks further demonstrate the applicability of adaptive processing; adaptive sampling provides an operating choice between fixed-sample configurations that balances credit stability and computational efficiency.

Overall, AdaSeq-MTA provides a practical approach to efficient multi-touch attribution for long journeys, enabling computational compression to account for both prediction quality and attribution changes. Attribution fidelity in this study primarily refers to consistency in model-based credit allocation, and the efficiency gains correspond to the evaluated data distribution and operating configurations.

Acknowledgements

The authors acknowledge Criteo for making the attribution dataset publicly available.

Statements and Declarations

Funding

The authors received no funding for this work.

Competing interests

The authors declare that they have no financial or non-financial competing interests relevant to this work.

Author contributions

Jinghui Wang: Conceptualization, Methodology, Software, Writing - original draft, Writing - review and editing. Haiyu Liu: Formal analysis, Writing - review and editing. Liyu Huang: Validation, Formal analysis, Visualization, Writing - review and editing.

Data availability

The original Criteo dataset used in this study is publicly available from its provider at https://ailab.criteo.com/criteo-attribution-modeling-bidding-dataset/. The processed datasets and synthetic data used in this study are not publicly released. These study-specific outputs have not been prepared and archived for public distribution.

Code availability

The code is not publicly released because the current implementation is tailored to the experimental workflow of this study, and preparing and maintaining a documented public release would require additional development resources.

Use of AI-assisted tools

During manuscript preparation, the authors used ChatGPT to assist with language editing, paragraph restructuring, and drafting revisions. The authors reviewed and revised the resulting text and take full responsibility for the final manuscript.

References

Abhishek V, Fader PS, Hosanagar K (2012) Media exposure through the funnel: A model of multi-stage attribution. Managerial Marketing eJournal URL https://api.semanticscholar.org/CorpusID:8444661

Adebayo J, Gilmer J, Muelly M, et al (2018) Sanity checks for saliency maps. In: Advances in Neural Information Processing Systems, URL https://arxiv.org/abs/1810.03292

Alvarez-Melis D, Jaakkola TS (2018) On the robustness of interpretability methods. In: ICML Workshop on Human Interpretability in Machine Learning, URL https://arxiv.org/abs/1806.08049

Banino A, Balaguer J, Blundell C (2021) PonderNet: Learning to ponder. arXiv preprint arXiv:2107.05407, URL https://arxiv.org/abs/2107.05407

Ben Mrad A, Hnich B (2024) Intelligent attribution modeling for enhanced digital marketing performance. Intelligent Systems with Applications 21:200337. https://doi.org/10.1016/j.iswa.2024.200337, URL https://doi.org/10.1016/j.iswa.2024.200337

Bencina J, Aykutlug E, Chen Y, et al (2025) LiDDA: Data driven attribution at LinkedIn. https://doi.org/10.48550/arXiv.2505.09861, URL https://arxiv.org/abs/2505.09861, arXiv:2505.09861

Cao Y, Zhou X, Feng J, et al (2022) Sampling is all you need on modeling long-term user behaviors for ctr prediction. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management. Association for Computing Machinery, New York, NY, USA, CIKM '22, pp 2974–2983, https://doi.org/10.1145/3511808.3557082, URL https://doi.org/10.1145/3511808.3557082

Chen Q, Pei C, Lv S, et al (2021) End-to-end user behavior retrieval in click-through rateprediction model. ArXiv abs/2108.04468. URL https://api.semanticscholar.org/CorpusID:236965682

Chen S, Chan Z, Sheng XR, et al (2025) See beyond a single view: Multi-attribution learning leads to better conversion rate prediction. In: Proceedings of the 34th ACM International Conference on Information and Knowledge Management, pp 5600–5608, https://doi.org/10.1145/3746252.3761580

Covert I, Lee SI (2021) Improving KernelSHAP: Practical shapley value estimation via linear regression. In: International Conference on Artificial Intelligence and Statistics, URL https://arxiv.org/abs/2012.01536

Covert I, Lundberg S, Lee SI (2020) Understanding global feature contributions with additive importance measures. In: Advances in Neural Information Processing Systems, URL https://arxiv.org/abs/2004.00668

Diemert E, Meynet J, Galland P, et al (2017) Attribution modeling increases efficiency of bidding in display advertising. In: Proceedings of the ADKDD'17. Association for Computing Machinery, New York, NY, USA, ADKDD'17, https://doi.org/10.1145/3124749.3124752, URL https://doi.org/10.1145/3124749.3124752

Graves A (2016) Adaptive computation time for recurrent neural networks. arXiv preprint arXiv:1603.08983, URL https://arxiv.org/abs/1603.08983

Guo C, Pleiss G, Sun Y, et al (2017a) On calibration of modern neural networks. In: International Conference on Machine Learning, URL https://arxiv.org/abs/1706.04599

Guo H, Tang R, Ye Y, et al (2017b) DeepFM: A factorization-machine based neural network for CTR prediction. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, pp 1725–1731, https://doi.org/10.24963/ijcai.2017/239, URL https://www.ijcai.org/proceedings/2017/239

Hinton G, Vinyals O, Dean J (2015) Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, URL https://arxiv.org/abs/1503.02531

Jain S, Wallace BC (2019) Attention is not explanation. In: Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, URL https://arxiv.org/abs/1902.10186

Jang E, Gu S, Poole B (2017) Categorical reparameterization with Gumbel-Softmax. In: International Conference on Learning Representations, URL https://arxiv.org/abs/1611.01144

Jethani N, Sudarshan M, Covert I, et al (2022) FastSHAP: Real-time shapley value estimation. In: International Conference on Learning Representations, URL https://arxiv.org/abs/2107.07436

Jin Y, Yang Y (2025) A survey on knowledge graph-based click-through rate prediction. Expert Systems with Applications 281:127501. https://doi.org/10.1016/j.eswa.2025.127501, URL https://doi.org/10.1016/j.eswa.2025.127501

Jordan P, Mahdian M, Vassilvitskii S, et al (2011) The multiple attribution problem in pay-per-conversion advertising. In: Proceedings of the 4th International Conference on Algorithmic Game Theory. Springer-Verlag, Berlin, Heidelberg, SAGT'11, pp 31–43

Jose A, Shetty SD (2022) Distilledctr: Accurate and scalable ctr prediction model through model distillation. Expert Systems with Applications 193:116474. https://doi.org/10.1016/j.eswa.2021.116474, URL https://www.sciencedirect.com/science/article/pii/S0957417421017541

Kingma DP, Ba J (2015) Adam: A method for stochastic optimization. In: International Conference on Learning Representations, URL https://arxiv.org/abs/1412.6980

Kumar S, Gupta G, Prasad R, et al (2020) Camta: Causal attention model for multi-touch attribution. In: 2020 International Conference on Data Mining Workshops (ICDMW), pp 79–86

Liu W, Zhou P, Zhao Z, et al (2020) FastBERT: A self-distilling BERT with adaptive inference time. In: Annual Meeting of the Association for Computational Linguistics, URL https://arxiv.org/abs/2004.02178

Lundberg SM, Lee SI (2017) A unified approach to interpreting model predictions. In: Advances in Neural Information Processing Systems, URL https://arxiv.org/abs/1705.07874

Ma JW, Zhao Z, Yi X, et al (2018) Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining URL https://api.semanticscholar.org/CorpusID:50770252

Maddison CJ, Mnih A, Teh YW (2017) The concrete distribution: A continuous relaxation of discrete random variables. In: International Conference on Learning Representations, URL https://arxiv.org/abs/1611.00712

Maurer A, Pontil M (2009) Empirical bernstein bounds and sample variance penalization. In: Proceedings of the 22nd Annual Conference on Learning Theory, URL https://www.learningtheory.org/colt2009/papers/012.pdf

Mitchell R, Cooper J, Frank E, et al (2022) Sampling permutations for shapley value estimation. Journal of Machine Learning Research 23(43):1–46. URL https://jmlr.org/papers/v23/21-0439.html

Park W, Kim D, Lu Y, et al (2019) Relational knowledge distillation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, URL https://openaccess.thecvf.com/content_CVPR_2019/html/Park_Relational_Knowledge_Distillation_CVPR_2019_paper.html

Pi Q, Bian W, Zhou G, et al (2019) Practice on long sequential user behavior modeling for click-through rate prediction. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. Association for Computing Machinery, New York, NY, USA, KDD '19, pp 2671–2679, https://doi.org/10.1145/3292500.3330666, URL https://doi.org/10.1145/3292500.3330666

Qi P, Zhu X, Zhou G, et al (2020) Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction. URL https://arxiv.org/abs/2006.05639, arXiv:2006.05639

Rao Y, Zhao W, Liu B, et al (2021) DynamicViT: Efficient vision transformers with dynamic token sparsification. In: Advances in Neural Information Processing Systems, pp 13937–13949, URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/747d3443e319a22747fbb873e8b2f9f2-Abstract.html

Raposo D, Ritter S, Richards B, et al (2024) Mixture-of-depths: Dynamically allocating compute in transformer-based language models. https://doi.org/10.48550/arXiv.2404.02258, URL https://arxiv.org/abs/2404.02258, arXiv:2404.02258

Ribeiro MT, Singh S, Guestrin C (2016) "why should i trust you?": Explaining the predictions of any classifier. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, URL https://arxiv.org/abs/1602.04938

Romero A, Ballas N, Kahou SE, et al (2015) FitNets: Hints for thin deep nets. In: International Conference on Learning Representations, URL https://arxiv.org/abs/1412.6550

Ross AS, Hughes MC, Doshi-Velez F (2017) Right for the right reasons: Training differentiable models by constraining their explanations. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, URL https://www.ijcai.org/proceedings/2017/0371.pdf

Rusthollkarhu S, Toukola S, Aarikka-Stenroos L, et al (2022) Managing B2B customer journeys in digital era: Four management activities with artificial intelligence-empowered tools. Industrial Marketing Management 104:241–257. https://doi.org/10.1016/j.indmarman.2022.04.014, URL https://doi.org/10.1016/j.indmarman.2022.04.014

Ryoo MS, Piergiovanni AJ, Arnab A, et al (2021) TokenLearner: Adaptive space-time tokenization for videos. In: Advances in Neural Information Processing Systems, pp 12786–12797, URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/6a30e32e56fce5cf381895dfe6ca7b6f-Abstract.html

Shapley LS (1952) A value for n-person games. Tech. Rep. P-295, RAND Corporation, https://doi.org/10.7249/P0295, URL https://www.rand.org/pubs/papers/P295.html

Shrikumar A, Greenside P, Kundaje A (2017) Learning important features through propagating activation differences. In: International Conference on Machine Learning, URL https://arxiv.org/abs/1704.02685

Singal R, Besbes O, Desir A, et al (2019) Shapley meets uniform: An axiomatic framework for attribution in online advertising. In: The World Wide Web Conference. Association for Computing Machinery, New York, NY, USA, WWW '19, pp 1713–1723, https://doi.org/10.1145/3308558.3313731, URL https://doi.org/10.1145/3308558.3313731

Sundararajan M, Taly A, Yan Q (2017) Axiomatic attribution for deep networks. In: International Conference on Machine Learning, URL https://arxiv.org/abs/1703.01365

Tang H, Liu J, Zhao M, et al (2020) Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations. In: Proceedings of the 14th ACM Conference on Recommender Systems. Association for Computing Machinery, New York, NY, USA, RecSys '20, pp 269–278, https://doi.org/10.1145/3383313.3412236, URL https://doi.org/10.1145/3383313.3412236

Tang J (2024) Dcrmta: Unbiased causal representation for multi-touch attribution. URL https://arxiv.org/abs/2401.08875v1, version 1, 16 January 2024, arXiv:2401.08875v1

Tung F, Mori G (2019) Similarity-preserving knowledge distillation. In: IEEE/CVF International Conference on Computer Vision, URL https://openaccess.thecvf.com/content_ICCV_2019/html/Tung_Similarity-Preserving_Knowledge_Distillation_ICCV_2019_paper.html

Wang X, Yu F, Dou ZY, et al (2018) SkipNet: Learning dynamic routing in convolutional networks. In: Proceedings of the European Conference on Computer Vision, pp 409–424, URL https://openaccess.thecvf.com/content_ECC

V_2018/html/Xin_Wang_SkipNet_Learning_Dynamic_ECCV_2018_paper.html

Wu J, Chen S, Chan Z, et al (2026) Mac: A conversion rate prediction benchmark featuring labels under multiple attribution mechanisms. In: Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2, pp 10009–10019, https://doi.org/10.1145/3770855.3817488, URL https://arxiv.org/abs/2603.02184v2

Wu S, Chen H, Quan X, et al (2023) ADKD: Attribution-driven knowledge distillation for language model compression. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp 8449–8465, https://doi.org/10.18653/v1/2023.acl-long.471, URL https://aclanthology.org/2023.acl-long.471/

Yang D, Dyer KP, Wang S (2020) Interpretable deep learning model for online multi-touch attribution. ArXiv abs/2004.00384. URL https://api.semanticscholar.org/CorpusID:214743433

Yang X, Li X, Liu Z, et al (2025) Multi-teacher knowledge distillation for debiasing recommendation with uniform data. Expert Systems with Applications 273:126808. https://doi.org/10.1016/j.eswa.2025.126808, URL https://doi.org/10.1016/j.eswa.2025.126808

Yang Y, Zhai P (2022) Click-through rate prediction in online advertising: A literature review. Information Processing & Management 59(2):102853. https://doi.org/10.1016/j.ipm.2021.102853, URL https://doi.org/10.1016/j.ipm.2021.102853

Yao D, Gong C, Zhang L, et al (2022) Causalmta: Eliminating the user confounding bias for causal multi-touch attribution. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. Association for Computing Machinery, New York, NY, USA, KDD '22, pp 4342–4352, https://doi.org/10.1145/3534678.3539108, URL https://doi.org/10.1145/3534678.3539108

Yeh CK, Hsieh CY, Suggala AS, et al (2019) On the (in)fidelity and sensitivity for explanations. In: Advances in Neural Information Processing Systems, URL https://arxiv.org/abs/1901.09392

Zagoruyko S, Komodakis N (2017) Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In: International Conference on Learning Representations, URL https://arxiv.org/abs/1612.03928

Zhou G, Mou N, Fan Y, et al (2018a) Deep interest evolution network for click-through rate prediction. URL https://arxiv.org/abs/1809.03672, arXiv:1809.03672

Zhou G, Song C, Zhu X, et al (2018b) Deep interest network for click-through rate prediction. URL https://arxiv.org/abs/1706.06978, arXiv:1706.06978