FROM PROSPECTIVE ACTIONS TO EVENTS: LIFECYCLE MARKED TEMPORAL POINT PROCESSES

Anonymous authors
Paper under double-blind review

ABSTRACT

A Marked Temporal Point Process (MTPP) is a stochastic process that models the timing and types of events in continuous time. MTPPs are widely used to characterize asynchronous event sequences in domains such as mobility, e-commerce, and online behavior. In many applications, event histories also contain prospective actions, such as destination searches or cart additions, that point to possible future targets but may never be realized. While existing models can represent these actions as part of event history, how an action's predictive status evolves as subsequent observations arrive has been under-explored. In this regime, we propose a prospective-action lifecycle MTPP (PAL-MTPP) that represents each prospective action as an evolving target-linked lifecycle state. PAL-MTPP updates these states using both subsequent events and no-event observations while preserving uncertainty about action realization. For events with multiple targets, we further introduce pairwise target interactions to refine target-set probabilities beyond independent target predictions. Experiments on real-world mobility and e-commerce datasets show that lifecycle-aware action modeling improves future-event prediction, while pairwise interactions provide additional gains for multi-target prediction.

1 INTRODUCTION

Event-time data are common in many real-world domains, including mobility (Wu et al., 2021; Hong et al., 2023), e-commerce (Xue et al., 2024), social interaction (Zhao et al., 2015; Du et al., 2015), and healthcare (Du et al., 2016). These data consist of asynchronous sequences of events with different types. A central task is to predict the time and type of future events from observed history. For example, previous visits may indicate the next destination, while shopping behavior may indicate when and what a user will purchase next. Such future events can be characterized by a probability distribution $f(t, m)$, where t denotes event time and m denotes event mark.

Temporal Point Processes (TPPs) provide a probabilistic framework for modeling event occurrence in continuous time (Rasmussen, 2018). Given an observed history $\mathcal{H}_\tau$, a TPP characterizes the next event time through the conditional distribution $f(t \mid \mathcal{H}_\tau)$. Marked Temporal Point Processes (MTPPs) extend TPPs by associating each event with a mark m that represents its type (Mei & Eisner, 2017). An MTPP models the joint distribution $f(t, m \mid \mathcal{H}_\tau)$ of the next event time and mark. Recent neural MTPPs use recurrent or attention-based architectures to capture complex temporal dependencies in observed event sequences (Mei & Eisner, 2017; Zhang et al., 2020; Zuo et al., 2020; Xue et al., 2024).

Observed histories in many applications contain more than realized events. They may also include prospective actions that indicate possible future targets. For example, destination searches and trip-planning records may indicate possible future visits in mobility (Jiang et al., 2022; Li & Shalaby, 2024), while cart additions may indicate possible future purchases in e-commerce (Esmeli & Gokce, 2025). Unlike realized events, prospective actions indicate possible rather than confirmed future outcomes. A searched destination may never be visited, and an added item may never be purchased.

A straightforward approach is to encode prospective actions as additional predictive features for future-event prediction (Li & Shalaby, 2024; Esmeli & Gokce, 2025). This preserves the action information but does not define how its predictive status changes after observation. A subsequent event involving the same target may indicate realization, but it may also occur independently of the

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

action. Conversely, the absence of events can also reduce the predictive value of an action over time. These observations motivate updating the predictive status of each action as new observations arrive.

The same applications may also produce events with multiple targets. In e-commerce, for example, individual cart additions act as prospective signals, while a subsequent purchase may contain several items in one basket. Such events can be represented by set-valued or multi-label marks (Chang et al., 2024; Shou et al., 2023). In this setting, lifecycle modeling determines whether individual prospective actions remain predictive, while target interactions determine which targets are likely to occur together. Independent target predictions do not capture these co-occurrence dependencies.

To address these challenges, we introduce a novel prospective-action lifecycle MTPP (PAL-MTPP). PAL-MTPP represents each prospective action as an evolving target-linked state. The state indicates whether the action remains pending and can still contribute to future-event prediction. PAL-MTPP updates lifecycle states using both subsequent events and no-event observations while preserving uncertainty about action realization. For multi-target events, PAL-MTPP further introduces pairwise target interactions to refine target-set probabilities beyond independent target predictions. The formalization of PAL-MTPP leads to the following contributions:

- We propose a novel MTPP modeling framework, i.e., PAL-MTPP, which represents prospective actions as evolving target-linked lifecycle states and updates their posterior from subsequent observations.
- We extend PAL-MTPP to multi-target events using pairwise target interactions that refine target-set probabilities beyond independent target predictions.
- We evaluate PAL-MTPP on real-world mobility and e-commerce datasets, showing improvements from lifecycle-aware action modeling and additional benefits from pairwise interactions for multi-target prediction.

2 Background

2.1 Temporal and Marked Temporal Point Processes

Temporal Point Process (TPP). A Temporal Point Process (TPP) models a sequence of ordered event times $\{t_n\}_{n=1}^{N}$ in continuous time. Given an observed history $\mathcal{H}_\tau$, the next event time is characterized by the conditional density $f(t \mid \mathcal{H}_\tau)$, with cumulative distribution $F(t \mid \mathcal{H}_\tau)$ and survival function $S(t \mid \mathcal{H}_\tau) = 1 - F(t \mid \mathcal{H}_\tau)$ (Rasmussen, 2018). The survival function gives the probability that no event occurs between τ and t. An equivalent representation uses the conditional intensity $\lambda(t \mid \mathcal{H}_t)$, which gives the instantaneous event rate at time t based on the history available at that time.

Marked Temporal Point Process (MTPP). A Marked Temporal Point Process extends a TPP by associating each event time t_n with a mark $m_n \in \mathcal{M}$ (Du et al., 2016; Mei & Eisner, 2017). The marked event is denoted by $e_n = (t_n, m_n)$, where the mark represents the event type or outcome. For a discrete mark space, let $\lambda_m(t \mid \mathcal{H}_t)$ denote the conditional intensity of mark $m \in \mathcal{M}$. The total event intensity is $\lambda(t \mid \mathcal{H}_t) = \sum_{m \in \mathcal{M}} \lambda_m(t \mid \mathcal{H}_t)$. Given a forecast origin τ, the joint density of the next event time $t > \tau$ and mark m is

$$f(t, m \mid \mathcal{H}_\tau) = \lambda_m(t \mid \mathcal{H}_t) \exp\left(-\int_\tau^t \lambda(u \mid \mathcal{H}_u)\, du\right). \tag{1}$$

This formulation jointly models when the next event occurs and which mark it takes (Daley & Vere-Jones, 2003).

2.2 Prospective Actions and Prediction Task

In addition to realized events, observed histories may contain prospective actions that indicate possible targets of future events. Let $\mathcal{H}_\tau$ contain observed marked events $e_n = (t_n, m_n)$ and prospective actions $a_i = (s_i, Q_i, X_i)$ with $s_i < \tau$. For action a_i, s_i denotes its observation time, Q_i is a distribution over possible future targets, and X_i contains auxiliary action attributes. When an action identifies a single target, Q_i becomes a point mass on that target. Depending on the application, the mark m_n may be a single target, such as a visited destination, or a set of targets, such as the products in a purchase basket.

Given a forecast origin τ, the task is to predict the time and mark of the next event using information available at τ. If no event is observed before the end of the observation window, the sequence is treated as right-censored. Periods without observed events also provide no-event observations for lifecycle updates and future prediction. Figure 1 illustrates the observation timeline and prediction task.

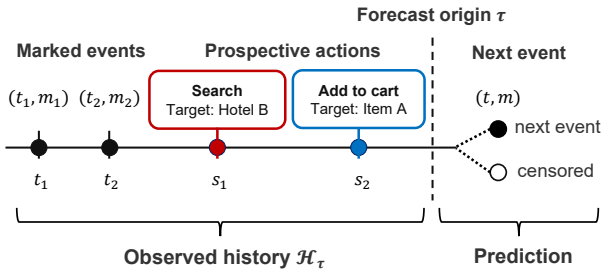


Figure 1: Observation timeline and prediction task.

3 Prospective-Action Lifecycle Marked Temporal Point Processes

Our approach focuses on modeling how the predictive status of prospective actions evolves after these actions are observed. Unlike approaches that encode actions as static historical information, PAL-MTPP represents each action with a lifecycle-aware state and uses this state to determine its contribution to future-event prediction. As shown in Figure 2, historical events provide baseline target-level intensity, while prospective actions provide lifecycle-aware target-specific effects. For multi-target events, pairwise interactions refine target composition while preserving the cardinality.

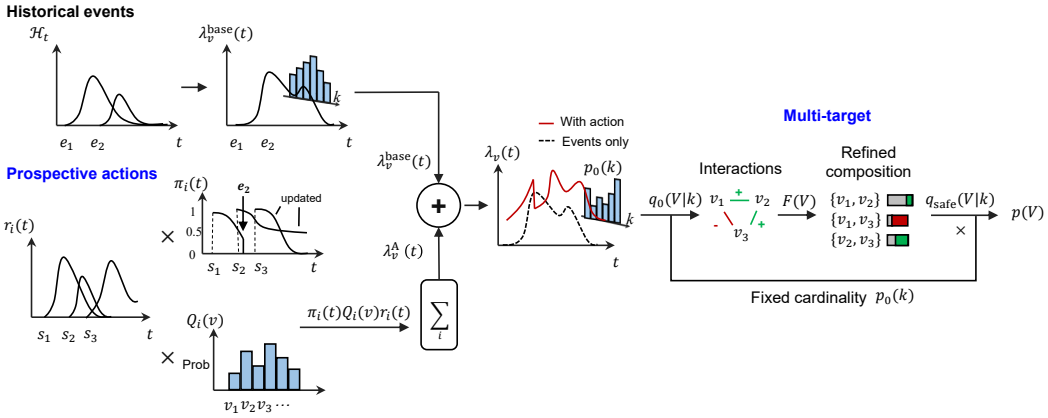


Figure 2: Overview of PAL-MTPP. Historical events provide baseline target-level intensity, while prospective actions contribute lifecycle-aware target-specific effects. Pairwise interactions refine target composition for multi-target events while preserving the predicted cardinality.

3.1 Target-Linked Lifecycle States

In this section, we introduce a target-linked lifecycle state to represent whether a prospective action remains pending. For a prospective action $a_i = (s_i, Q_i, X_i)$ observed at time s_i, PAL-MTPP associates a binary lifecycle state $Z_i(t) \in \{0, 1\}$ for $t \geq s_i$. The state is initialized as $Z_i(s_i) = 1$. A value of one indicates that the action remains pending and can still contribute to future-event prediction, while a value of zero indicates that it is no longer pending. A pending action may leave this state through realization, cancellation, or hidden expiry. Realization corresponds to the occurrence of the intended future event, cancellation to an explicit removal or invalidation of the action, and hidden expiry to an unobserved transition out of the pending state over time.

3.2 Lifecycle-Conditioned Event Intensity

In PAL-MTPP, the target-level event intensity is defined as the sum of a baseline intensity and contributions from pending prospective actions. Each action is associated with a target distribution Q_i,

which determines the targets it can affect. A lifecycle configuration is denoted by $z = (z_1, \ldots, z_L)$, where L is the number of represented prospective actions and z_i is a realization of the lifecycle state $Z_i(t)$. A value $z_i = 1$ indicates that action a_i remains pending. Conditioned on configuration z, the intensity for target $v \in \mathcal{V}$ is

$$\lambda_v(t, z) = \lambda_v^{\text{base}}(t) + \sum_{i:s_i<t} z_i Q_i(v) g_i(t - s_i), \tag{2}$$

where $\lambda_v^{\text{base}}(t)$ is the baseline intensity for target v, $Q_i(v)$ is the target distribution of action a_i, and $g_i(t - s_i)$ determines the action contribution as a function of its age. Since Q_i is normalized over the target space, we define $r_i(t) = g_i(t - s_i)$ as the realization rate of action a_i conditional on the action remaining pending. Thus, $r_i(t)$ controls the total action contribution, while $Q_i(v)$ distributes this contribution across possible targets.

3.3 Lifecycle Updates from Observations

associated The lifecycle states are not directly observed. Exact filtering over L prospective actions requires a posterior over 2^L lifecycle configurations. PAL-MTPP instead uses a factorized approximation and maintains only the marginal pending probability $\pi_i(t)$ of each action:

$$p_t(z) \approx \prod_{i=1}^{L} \pi_i(t)^{z_i} \big(1 - \pi_i(t)\big)^{1-z_i}, \tag{3}$$

where $\pi_i(t)$ is the posterior probability that action a_i remains pending. This reduces lifecycle inference to one probability per action. Both observed events and periods without events update these probabilities. Consider an interval of length Δ during which the realization rate r_i and hidden-expiry rate ξ_i are constant. Let $S_i(\Delta)$ denote the probability that no realization of action a_i is observed during this interval. The no-event update is

$$\begin{aligned} S_i(\Delta) &= 1 - \frac{\pi_i(t) r_i}{r_i + \xi_i} \left(1 - e^{-(r_i+\xi_i)\Delta}\right), \\ \pi_i(t + \Delta) &= \frac{\pi_i(t) e^{-(r_i+\xi_i)\Delta}}{S_i(\Delta)}. \end{aligned} \tag{4}$$

For constant rates, this recursion is equivalent to

$$\frac{d\pi_i(t)}{dt} = -\xi_i(t)\pi_i(t) - r_i(t)\pi_i(t)\big(1 - \pi_i(t)\big). \tag{5}$$

The first term accounts for hidden expiry, while the second captures the effect of observing no realization. When the rates vary with action age, the update is applied piecewise across rate intervals. Let t^- and t^+ denote the moments immediately before and after an observed event at time t. When an event associated with target v is observed at time t, it may come from the baseline process or from the realization of a pending action.

PAL-MTPP updates the marginal pending probabilities by marginalizing over these possible event sources rather than treating every target-matching event as a deterministic realization. Explicit cancellation observations remove the corresponding action from the pending state. Event observations can induce dependencies among actions. PAL-MTPP does not retain these dependencies; after each event update, it keeps only the updated marginals $\{\pi_i(t^+)\}$ and continues filtering under the factorized approximation.

3.4 Prediction and Learning

Under the factorized lifecycle posterior, each prospective action contributes to future-event prediction according to its probability of remaining pending. The predictive intensity for target v is

$$\widehat{\lambda}_v(t \mid \mathcal{H}_{t^-}) = \lambda_v^{\text{base}}(t) + \sum_{i:s_i<t} \pi_i(t^-)\, Q_i(v)\, r_i(t). \tag{6}$$

The first term gives the baseline intensity from event history. In the second term, $\pi_i(t^-)$ is the posterior probability that action a_i remains pending, $Q_i(v)$ determines which targets it supports,

and $r_i(t)$ determines its realization rate at time t. For single-target events, the total event intensity is $\lambda_{\text{event}}(t \mid \mathcal{H}_{t^-}) = \sum_{v \in \mathcal{V}} \widehat{\lambda}_v(t \mid \mathcal{H}_{t^-})$, and the conditional mark distribution is induced by the target-specific intensities,

$$p_\theta(v \mid t, \mathcal{H}_{t^-}) = \frac{\widehat{\lambda}_{v,\theta}(t \mid \mathcal{H}_{t^-})}{\lambda_{\text{event},\theta}(t \mid \mathcal{H}_{t^-})}, \tag{7}$$

where θ denotes the learnable parameters of the lifecycle event model. The event-time and mark terms are two components of the same marked event likelihood. The model is trained using the standard marked temporal point process likelihood (Daley & Vere-Jones, 2003; Mei & Eisner, 2017). Over an observation window $[0, T]$, the negative log-likelihood (NLL) is

$$\begin{aligned} \mathcal{L}_{\text{joint}}(\theta) = & \int_0^T \lambda_{\text{event},\theta}(t \mid \mathcal{H}_{t^-})\, dt - \sum_{n=1}^{N} \log \lambda_{\text{event},\theta}(t_n \mid \mathcal{H}_{t_n^-}) \\ & - \sum_{n=1}^{N} \log p_\theta(m_n \mid t_n, \mathcal{H}_{t_n^-}). \end{aligned} \tag{8}$$

The integral term accounts for intervals without observed events, including the right-censored interval after the last event. The remaining terms assign high likelihood to the observed event times and marks. For multi-target events, the event-time and target-set models are factorized. Section 4 defines the target-set distribution and its interaction refinement.

4 Pairwise Interactions for Multi-Target Events

The lifecycle model in Section 3 provides target-level predictions for future events. For multi-target events, we use these predictions as the baseline for a second-stage target-composition refinement. In mobility, an event usually corresponds to a single destination, while a purchase event in e-commerce may contain multiple items. For multi-target events, individual target predictions alone do not capture dependencies among targets that occur together. Therefore, we model the target-set distribution by separating event cardinality from target composition, and use pairwise interactions to refine the composition distribution while preserving the predicted cardinality.

4.1 Target-Set Distribution

For multi-target events, prediction involves both how many targets occur and which targets occur together (Chang et al., 2024; Shou et al., 2023). The lifecycle model provides target-level predictions to a set prediction head, which constructs a baseline target-set distribution before interaction refinement. For each candidate target v, the set prediction head outputs a logit ℓ_v, with Bernoulli parameter $\rho_v = \sigma(\ell_v)$. Assuming independent target selections, the baseline distribution over target sets is

$$p_0(V) = \frac{\prod_{v \in V} \rho_v \prod_{u \in \mathcal{V} \setminus V} (1 - \rho_u)}{1 - \prod_{u \in \mathcal{V}} (1 - \rho_u)}, \qquad V \neq \varnothing. \tag{9}$$

The denominator removes the probability mass assigned to the empty set. Let $V \subseteq \mathcal{V}$ denote a nonempty target set and let $k = |V|$ denote its cardinality. We decompose the baseline target-set distribution as

$$p_0(V) = p_0(k)\, q_0(V \mid k), \tag{10}$$

where $p_0(k) = \sum_{U:|U|=k} p_0(U)$ is the cardinality distribution, and $q_0(V \mid k) = \frac{p_0(V)}{\sum_{U:|U|=k} p_0(U)}$ is the conditional composition distribution given cardinality k. This decomposition separates event size from target composition. The interaction module keeps the cardinality distribution $p_0(k)$ unchanged and refines only the composition distribution $q_0(V \mid k)$, so pairwise interactions model co-occurrence dependencies without changing the predicted event size.

4.2 Pairwise Interaction Refinement

Targets in a multi-target event may exhibit positive or negative dependencies (Wang & Cao, 2021). To model these dependencies without considering all pairs in the target space, we apply interaction

refinement to a focal subset of candidate targets. Let $\mathcal{F} \subseteq \mathcal{V}$ denote the focal target set. For a target set V, we define its interaction score as

$$F(V) = \sum_{\{u,v\} \subseteq V \cap \mathcal{F}} \beta_{uv}, \tag{11}$$

where β_{uv} represents the pairwise interaction between targets u and v. A positive value increases the relative probability of target sets containing the pair, while a negative value decreases it. Targets outside $\mathcal{F}$ remain in the baseline target-set distribution without additional interaction scores.

Learning a separate parameter for every target pair scales poorly with the number of targets. We instead parameterize the interactions using shared target representations. Let x_v denote the feature representation of target v. We compute

$$\begin{aligned} h_v &= \tanh(W x_v + b), \\ \beta_{uv} &= 2 \tanh\left(\frac{h_u^\top \operatorname{diag}(\omega) h_v}{2\sqrt{d}} \right), \end{aligned} \tag{12}$$

where W, b, and ω are shared learnable parameters and d is the dimension of h_v. The same interaction function is shared across all focal target pairs. We refine the baseline composition distribution using the interaction score $F(V)$:

$$q_F(V \mid k) = \frac{q_0(V \mid k) \exp(F(V))}{\sum_{V' : |V'| = k} q_0(V' \mid k) \exp(F(V'))}. \tag{13}$$

The exponential weighting increases or decreases the relative probability of a target set according to its interaction score. Normalization is restricted to sets with the same cardinality k, so the refinement changes target composition while leaving $p_0(k)$ unchanged. For $k = 1$, no target pair exists and $F(V) = 0$, giving $q_F(V \mid 1) = q_0(V \mid 1)$. To limit the deviation from the baseline distribution, we combine the baseline and refined composition distributions:

$$q_{\text{safe}}(V \mid k) = (1 - \alpha_k) q_0(V \mid k) + \alpha_k q_F(V \mid k), \tag{14}$$

where $\alpha_k \in [0, 1]$ controls the refinement strength. The final target-set distribution is $p(V) = p_0(k)\, q_{\text{safe}}(V \mid k)$.

4.3 Learning Objective

The baseline target-set predictor is kept fixed while the interaction module is trained. This second-stage training isolates the effect of target interactions and preserves the baseline event-time, target, and cardinality predictions. For an observed target set V with cardinality k, we optimize the refined composition distribution while regularizing its deviation from the baseline. The objective is

$$\mathcal{L}_{\text{int}} = -\log q_F(V \mid k) + \lambda_{\text{KL}} D_{\text{KL}} \left(q_0(\cdot \mid k) \| q_F(\cdot \mid k) \right), \tag{15}$$

where λ_{KL} controls the regularization strength. The first term increases the probability of the observed target set, while the KL term keeps the refined composition distribution close to the baseline. During training, the interaction parameters are optimized using $q_F(V \mid k)$. During prediction, the safe mixture in Eq. 14 controls the strength of the interaction refinement.

5 Related Work

Neural TPP modeling. Neural temporal point processes extend classical point-process models with learned representations of event history. Early methods used recurrent networks to parameterize conditional intensities (Du et al., 2016; Mei & Eisner, 2017), while later approaches adopted self-attention and Transformer architectures to capture longer-range temporal dependencies (Zhang et al., 2020; Zuo et al., 2020; Yang et al., 2022). Other work has improved temporal flexibility through neural intensity parameterization, direct modeling of inter-event distributions, and continuous-time latent dynamics (Omi et al., 2019; Shchur et al., 2020; Chen et al., 2021; Chang et al., 2025). These approaches primarily model how observed event history influences future events. Our setting additionally considers prospective actions whose predictive status can change before realization.

Prospective actions. Observed histories may contain prospective actions that refer to possible future outcomes. In mobility, trip-planning and search records have been used to study information-seeking behavior and its relation to later physical visits (Li & Shalaby, 2024; Jiang et al., 2022). In e-commerce, cart additions provide similar information about possible future purchases (Esmeli & Gokce, 2025). These actions can be predictive, but their realization remains uncertain. A searched destination may be visited after intervening events or may never be visited, while an added item may later be purchased or remain unpurchased. Existing studies mainly relate such actions to later outcomes or incorporate them as additional predictive features. They do not explicitly model how the status of an individual prospective action changes as subsequent observations arrive.

Multi-target event modeling. Some event streams contain multiple targets within the same event, such as product baskets or other set-valued outcomes. Set-valued point-process models extend conventional MTPPs from single marks to sets of items (Chang et al., 2024), while concurrent multi-label models capture dependencies among event types that occur together (Shou et al., 2023). Related work on next-basket recommendation also models dependencies among items within the same basket and across consecutive baskets (Wang & Cao, 2021). These studies motivate modeling dependencies among targets rather than treating them independently. In our setting, pairwise interactions refine the composition of a predicted target set while preserving its predicted cardinality.

6 EXPERIMENTS

We evaluate PAL-MTPP on real-world mobility and e-commerce datasets from three perspectives. We first compare overall future-event prediction performance with conventional neural TPP and action-based baselines (Section 6.2). We then analyze whether lifecycle updates adjust prospective-action contributions as new observations arrive (Section 6.3). Finally, we evaluate whether pairwise interactions improve prediction for multi-target events (Section 6.4).

6.1 EXPERIMENTAL SETUP

Datasets. We evaluate PAL-MTPP on three datasets from mobility and e-commerce domains. LBS-IntentBench contains mobility histories and destination-related actions, where the task is to predict the time and destination of the next stay. Synerise and Retailrocket contain e-commerce interactions with purchase events represented as target sets. In both e-commerce datasets, cart additions are treated as prospective actions, while purchases or transactions provide realization observations for lifecycle updates.

Metrics. We use Joint NLL as the main metric for joint event-time and mark prediction, where lower values indicate better probabilistic prediction. For mobility, we additionally report Target NLL, which evaluates the conditional distribution over destinations, and Top-1 accuracy, which measures whether the highest-probability destination matches the observed one. For e-commerce, we report Set NLL and NDCG@10 to evaluate the predicted target-set distribution and ranking quality.

Baselines. We compare PAL-MTPP with three representative neural TPP models: Neural Hawkes Process (NHP; Mei & Eisner, 2017), Transformer Hawkes Process (THP; Zuo et al., 2020), and S2P2 (Chang et al., 2025). NHP models event history with a recurrent continuous-time hidden state, THP uses self-attention to capture long-range temporal dependencies, and S2P2 models continuous-time event dynamics with a state-space architecture. We also include two action-based baselines to separate the effect of lifecycle modeling from simply incorporating prospective-action information. Action-Feature aggregates action history into fixed features, while Action-Transformer encodes action records with an attention-based representation.

6.2 OVERALL PREDICTION PERFORMANCE

Table 1 summarizes the overall prediction results. PAL-MTPP achieves the lowest Joint NLL on all three datasets, outperforming both conventional neural TPP models and action-based baselines. On LBS-IntentBench, PAL-MTPP also achieves the lowest Target NLL and the highest Top-1 accuracy. On Synerise and Retailrocket, it obtains the lowest Set NLL, with the best NDCG@10 on Synerise and a tie with Action-Transformer on Retailrocket. These results indicate that lifecycle-aware

Table 1: Overall prediction performance on mobility and e-commerce datasets.

(a) Mobility	**LBS-IntentBench**		
Method	Joint NLL ↓	Target NLL ↓	Top-1 ↑
NHP	3.989	1.680	40.94
THP	3.999	1.685	40.79
S2P2	5.057	1.753	38.84
Action-Feature	3.950	1.627	42.52
Action-Transformer	3.982	1.653	41.60
PAL-MTPP	**3.794**	**1.589**	**45.13**

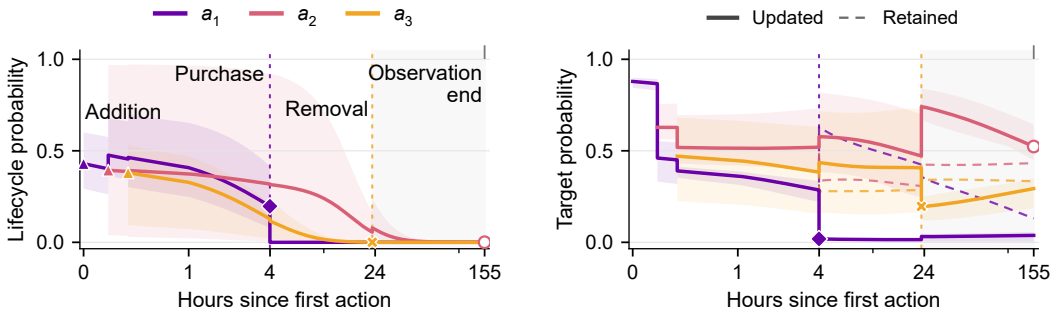
(b) E-commerce	**Synerise**			**Retailrocket**		
Method	Joint NLL ↓	Set NLL ↓	NDCG@10 ↑	Joint NLL ↓	Set NLL ↓	NDCG@10 ↑
NHP	0.414	1.642	87.12	0.232	1.059	95.15
THP	0.423	1.695	87.07	0.229	1.002	95.16
S2P2	0.405	1.649	86.85	0.235	1.116	94.66
Action-Feature	0.425	1.717	86.84	0.229	1.014	95.47
Action-Transformer	0.424	1.670	87.07	0.224	0.920	**96.04**
PAL-MTPP	**0.403**	**1.521**	**88.03**	**0.222**	**0.889**	**96.04**

Note. Joint NLL includes event-time and mark likelihoods. Top-1 and NDCG@10 are reported as percentages. Bold indicates the best result in each column, including ties.

prospective-action modeling improves both joint future-event prediction and target prediction across mobility and e-commerce.

6.3 Lifecycle Update Analysis

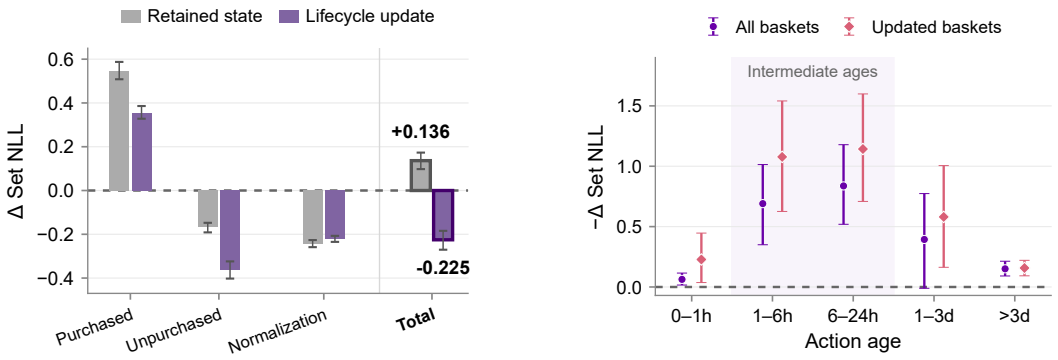
We first examine how lifecycle updates change the contribution of prospective actions as new observations arrive. Figure 3 shows three cart actions from a Synerise sequence. Their lifecycle probabilities evolve over time, and purchase or removal observations reduce the probability that the corresponding actions remain pending. These changes also affect the predicted target probabilities. In contrast, retaining previous lifecycle states continues to assign probability to targets associated with completed or removed actions. This comparison isolates the effect of lifecycle updating from the action representation itself.



(a) Lifecycle probability (b) Target prediction

Figure 3: Lifecycle dynamics on Synerise. (a) Lifecycle probabilities for three prospective actions. (b) Target probabilities under updated (solid) and retained (dashed) states. Lines show the mean across three seeds and bands show their range.

We next examine the effect of lifecycle updates at the population level. Figure 4(a) shows the change in Set NLL under different lifecycle-update settings. Completion-aware updates reduce Set NLL by 0.225, whereas applying the same update to retention-trained models increases Set NLL by 0.136. Figure 4(b) shows that the improvement is largest for actions of intermediate age. This suggests that lifecycle updates are most useful when an action remains predictive but its realization status becomes more uncertain over time.



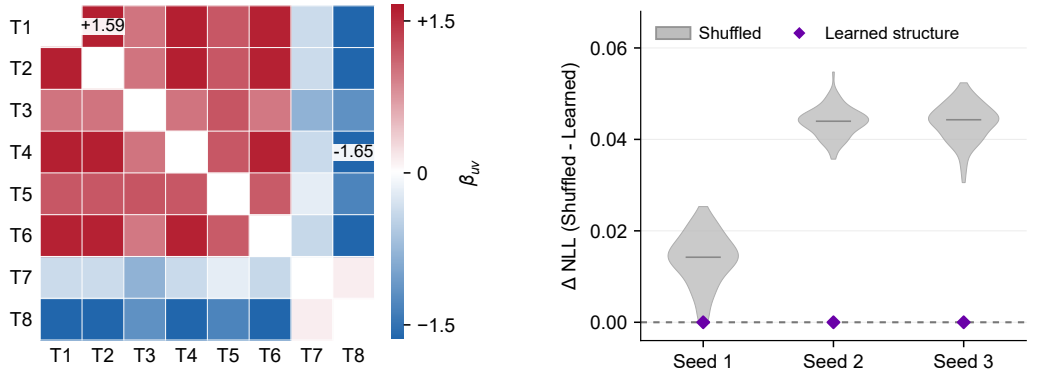
(a) Set NLL change

(b) Gain across action ages

Figure 4: Lifecycle analysis on Synerise. (a) Change in Set NLL under different lifecycle-update settings. (b) Improvement across action ages. Error bars show 95% bootstrap intervals.

6.4 Multi-Target Interaction Analysis

We examine whether the learned pairwise interactions capture target-specific dependencies. Figure 5(a) shows a learned interaction matrix from Synerise, where positive and negative coefficients increase or decrease the relative probability of target sets containing the corresponding pairs. To test whether prediction depends on the learned pair structure rather than only on the interaction strengths, we shuffle the target-pair assignments while keeping the coefficient values fixed. As shown in Figure 5(b), shuffling increases validation NLL by 0.0337 on average across three seeds. This indicates that prediction depends on the learned pair structure, not only on the overall distribution of interaction strengths. Pairwise interactions also outperform additive target corrections, indicating additional value from modeling target combinations beyond independent target-level adjustments.



(a) Learned signed pairwise interactions

(b) NLL change under pair-assignment shuffling

Figure 5: Multi-target interaction analysis on Synerise. (a) Learned signed pairwise interaction coefficients. (b) Change in validation NLL after shuffling target-pair assignments while keeping coefficient values fixed. Positive values indicate higher validation NLL after shuffling.

7 Conclusion

In this work, we presented PAL-MTPP, a novel MTPP for modeling prospective actions whose predictive status evolves over time. PAL-MTPP uses target-linked lifecycle states updated by event and no-event observations, and pairwise interactions to refine multi-target composition while preserving predicted cardinality. Experiments show that lifecycle-aware action modeling improves future-event prediction, with the largest gains for actions of intermediate age. Pairwise interaction structure also contributes to multi-target prediction, as shuffling learned target-pair assignments degrades validation likelihood.

AI USE STATEMENT

We used OpenAI Codex to assist with code implementation, data processing, manuscript drafting, language polishing, and formatting, as well as figure presentation and bibliographic verification. The authors reviewed the AI-assisted work and take responsibility for the final content of the paper, including its code, data processing, analyses, and claims.

ETHICS STATEMENT

We use publicly released processed LBS-IntentBench data and the Synerise and Retailrocket datasets under their respective CC BY-NC 4.0, CC BY-NC 4.0, and CC BY-NC-SA 4.0 licenses. The LBS-IntentBench release excludes raw mobility records and private identifier mappings; Synerise and Retailrocket report anonymization and hashing. Applications should protect user privacy and avoid unauthorized tracking, profiling, or discriminatory use.

REFERENCES

Yuxin Chang, Alex J. Boyd, and Padhraic Smyth. Probabilistic modeling for sequences of sets in continuous-time. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 4357–4365. PMLR, 2024. URL https://proceedings.mlr.press/v238/chang24a.html.

Yuxin Chang, Alex Boyd, Cao Xiao, Taha Kass-Hout, Parminder Bhatia, Padhraic Smyth, and Andrew Warrington. Deep continuous-time state-space models for marked event sequences. In *Advances in Neural Information Processing Systems*, volume 38, pp. 162672–162713, 2025. doi: 10.52202/085713-5431. URL https://papers.nips.cc/paper_files/paper/2025/hash/ee39348acc798915d2d15a8bbbd417b8-Abstract-Conference.html.

Ricky T. Q. Chen, Brandon Amos, and Maximilian Nickel. Neural spatio-temporal point processes. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=XQQA6-So14.

D. J. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Processes: Volume I: Elementary Theory and Methods*. Springer, 2nd edition, 2003. doi: 10.1007/b97277. URL https://doi.org/10.1007/b97277.

Nan Du, Mehrdad Farajtabar, Amr Ahmed, Alexander J. Smola, and Le Song. Dirichlet-Hawkes processes with applications to clustering continuous-time document streams. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 219–228, 2015. doi: 10.1145/2783258.2783411. URL https://doi.org/10.1145/2783258.2783411.

Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1555–1564, 2016. doi: 10.1145/2939672.2939875. URL https://doi.org/10.1145/2939672.2939875.

Ramazan Esmeli and Aytac Gokce. An analysis of consumer purchase behavior following cart addition in e-commerce utilizing explainable artificial intelligence. *Journal of Theoretical and Applied Electronic Commerce Research*, 20(1):28, 2025. doi: 10.3390/jtaer20010028. URL https://doi.org/10.3390/jtaer20010028.

Ye Hong, Yatao Zhang, Konrad Schindler, and Martin Raubal. Context-aware multi-head self-attentional neural network model for next location prediction. *Transportation Research Part C: Emerging Technologies*, 156:104315, 2023. doi: 10.1016/j.trc.2023.104315. URL https://doi.org/10.1016/j.trc.2023.104315.

Renhe Jiang, Quanjun Chen, Zekun Cai, Zipei Fan, Xuan Song, Kota Tsubouchi, and Ryosuke Shibasaki. Will you go where you search? A deep learning framework for estimating user search-and-go behavior. *Neurocomputing*, 472:338–348, 2022. doi: 10.1016/j.neucom.2020.10.001. URL https://doi.org/10.1016/j.neucom.2020.10.001.

Lisa Li and Amer Shalaby. Navigating the transit network: Understanding riders' information seeking behavior using trip planning data. *Transportation Research Part A: Policy and Practice*, 185: 104096, 2024. doi: 10.1016/j.tra.2024.104096. URL https://doi.org/10.1016/j.tra.2024.104096.

Hongyuan Mei and Jason Eisner. The neural Hawkes process: A neurally self-modulating multivariate point process. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/6463c88460bd63bbe256e495c63aa40b-Abstract.html.

Takahiro Omi, Naonori Ueda, and Kazuyuki Aihara. Fully neural network based model for general temporal point processes. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/39e4973ba3321b80f37d9b55f63ed8b8-Abstract.html.

Jakob Gulddahl Rasmussen. Lecture notes: Temporal point processes and the conditional intensity function. arXiv preprint arXiv:1806.00221, 2018. URL https://arxiv.org/abs/1806.00221.

Oleksandr Shchur, Marin Biloš, and Stephan Günnemann. Intensity-free learning of temporal point processes. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HygOjhEYDH.

Xiao Shou, Tian Gao, Dharmashankar Subramanian, Debarun Bhattacharjya, and Kristin P. Bennett. Concurrent multi-label prediction in event streams. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 9820–9828, 2023. doi: 10.1609/aaai.v37i8.26172. URL https://doi.org/10.1609/aaai.v37i8.26172.

Wei Wang and Longbing Cao. Interactive sequential basket recommendation by learning basket couplings and positive/negative feedback. *ACM Transactions on Information Systems*, 39(3):1–26, 2021. doi: 10.1145/3444368. URL https://doi.org/10.1145/3444368. Article 24.

Yuankai Wu, Zhanhong Cheng, and Lijun Sun. Individual mobility prediction via attentive marked temporal point processes. arXiv preprint arXiv:2109.02715, 2021. URL https://arxiv.org/abs/2109.02715.

Siqiao Xue, Xiaoming Shi, Zhixuan Chu, Yan Wang, Hongyan Hao, Fan Zhou, Caigao Jiang, Chen Pan, James Y. Zhang, Qingsong Wen, Jun Zhou, and Hongyuan Mei. EasyTPP: Towards open benchmarking temporal point processes. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=PJwAkg0z7h.

Chenghao Yang, Hongyuan Mei, and Jason Eisner. Transformer embeddings of irregularly spaced events and their participants. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Rty5g9imm7H.

Qiang Zhang, Aldo Lipani, Omer Kirnap, and Emine Yilmaz. Self-attentive Hawkes process. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 11183–11193. PMLR, 2020. URL https://proceedings.mlr.press/v119/zhang20q.html.

Qingyuan Zhao, Murat A. Erdogdu, Hera Y. He, Anand Rajaraman, and Jure Leskovec. SEISMIC: A self-exciting point process model for predicting tweet popularity. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1513–1522, 2015. doi: 10.1145/2783258.2783401. URL https://doi.org/10.1145/2783258.2783401.

Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. Transformer Hawkes process. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 11692–11702. PMLR, 2020. URL https://proceedings.mlr.press/v119/zuo20a.html.

A NOTATION

Table 2 summarizes the main notation used in PAL-MTPP.

Table 2: Main notation used in PAL-MTPP.

Symbol	Meaning
Events and targets	
$e_n = (t_n, m_n)$	Observed event n with time t_n and mark $m_n \in \mathcal{M}$
$\mathcal{H}_t$	Observed history available at time t
τ	Forecast origin
$v \in \mathcal{V}$	Target v in target space $\mathcal{V}$
$V_n \subseteq \mathcal{V},\ k_n = \lvert V_n \rvert$	Target set of event e_n and its cardinality
Prospective actions and lifecycle states	
$a_i = (s_i, Q_i, X_i)$	Prospective action i with observation time s_i, target distribution Q_i, and attributes X_i
$Z_i(t) \in \{0, 1\}$	Lifecycle state of action a_i; 1 denotes pending and 0 denotes inactive
$z = (z_1, \ldots, z_L)$	One possible lifecycle configuration over the represented actions
$r_i(t)$, $\xi_i(t)$	Realization and hidden-expiry rates
$\pi_i(t)$	Posterior probability that action a_i remains pending
Lifecycle inference and prediction	
$p_t(z)$	Factorized approximate posterior over lifecycle configurations
$\lambda_v^{\text{base}}(t)$	Baseline intensity for target v
$\lambda_v(t, z)$	Target-level intensity under lifecycle configuration z
$\widehat{\lambda}_v(t \mid \mathcal{H}_{t^-})$	Predictive intensity for target v
Multi-target interaction	
$V \subseteq \mathcal{V},\ k = \lvert V \rvert$	Generic target set and its cardinality
ℓ_v, ρ_v	Target logit and Bernoulli parameter
$p_0(V)$, $p_0(k)$, $q_0(V \mid k)$	Baseline target-set, cardinality, and composition distributions
$\mathcal{F} \subseteq \mathcal{V}$	Focal subset of candidate targets
x_v, β_{uv}	Target feature and signed interaction coefficient
$F(V)$	Pairwise interaction score of target set V
$q_F(V \mid k)$	Interaction-refined composition distribution
α_k, $q_{\text{safe}}(V \mid k)$	Refinement weight and final composition distribution

B ADDITIONAL EXPERIMENTS

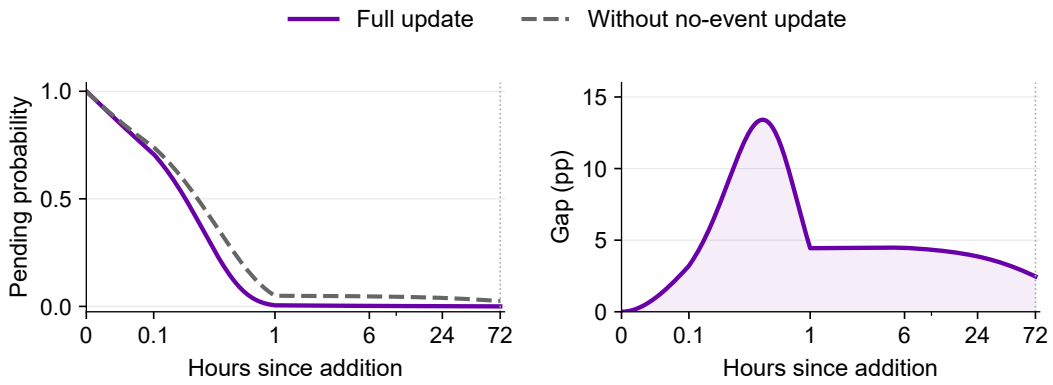
Additional experiments are conducted to further analyze the mechanisms of PAL-MTPP.

B.1 EFFECT OF NO-EVENT UPDATES

In Section 3.3, PAL-MTPP updates lifecycle states using both event occurrences and periods without observed events. To examine the role of no-event observations, we fix the model parameters and compare lifecycle inference with and without no-event updates. Figure 6 shows the evolution of the pending probability under the two settings. Without no-event updates, the posterior only changes when an event is observed. With the complete update mechanism, the posterior also evolves during periods without events. This comparison illustrates how event absence contributes to lifecycle inference in PAL-MTPP.

B.2 TARGET-SET INTERACTION REFINEMENT

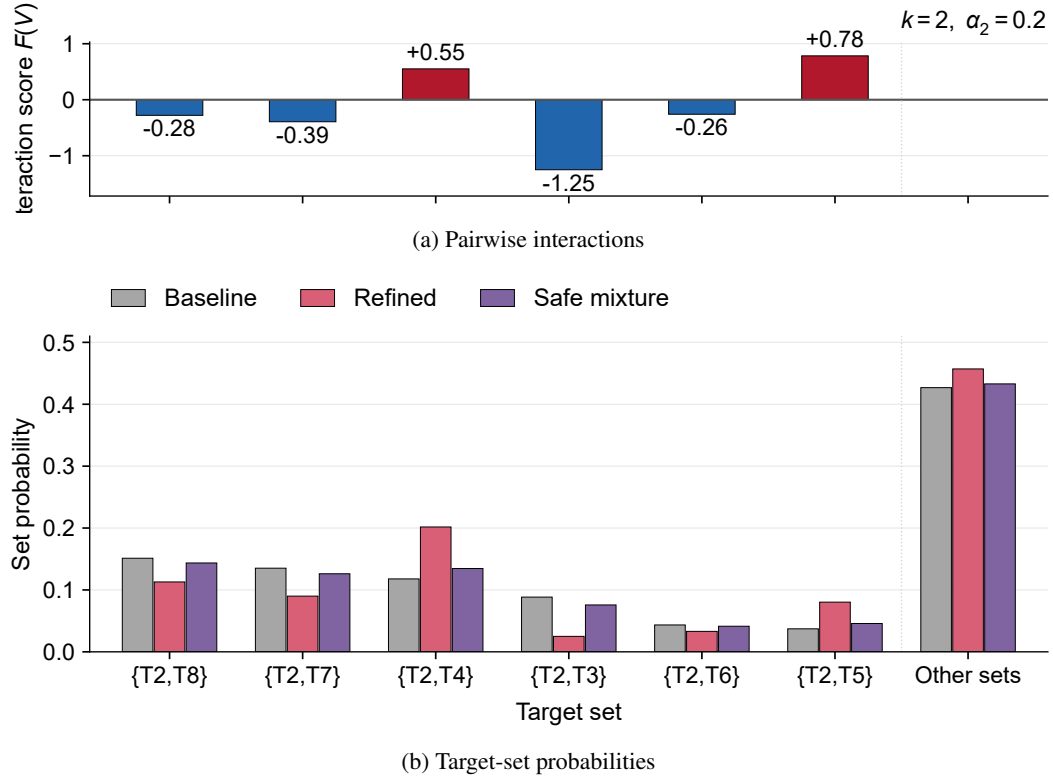
In Section 4, PAL-MTPP introduces pairwise interactions to refine target combinations while preserving the predicted event cardinality. Figure 7 compares the baseline target-set distribution $q_0(V|k)$, the interaction-refined distribution $q_F(V|k)$, and the final safe mixture $q_{\text{safe}}(V|k)$. The interaction refinement changes the relative probabilities of target combinations. Positive interactions increase the probability of target sets containing the corresponding pairs, while negative interactions reduce their probability. The safe mixture controls the deviation from the baseline distribution.



(a) Lifecycle posterior

(b) Probability gap

Figure 6: Effect of no-event updates on lifecycle posterior evolution. The comparison disables no-event updates. The gap is the control minus the full-update posterior.



(a) Pairwise interactions

(b) Target-set probabilities

Figure 7: Target-set probability refinement with pairwise interactions. The figure compares the baseline distribution $q_0(V|k)$, the interaction-refined distribution $q_F(V|k)$, and the final safe mixture $q_{\text{safe}}(V|k)$.

B.3 Synthetic Intensity Recovery

In Section 3.2, PAL-MTPP models event intensity by combining baseline intensity with lifecycle-aware prospective-action contributions. Since the true intensity is unavailable in real-world datasets, we conduct a synthetic experiment where the ground-truth intensity is available.

We compare three variants:

(1) Event-only, which does not model prospective-action effects;

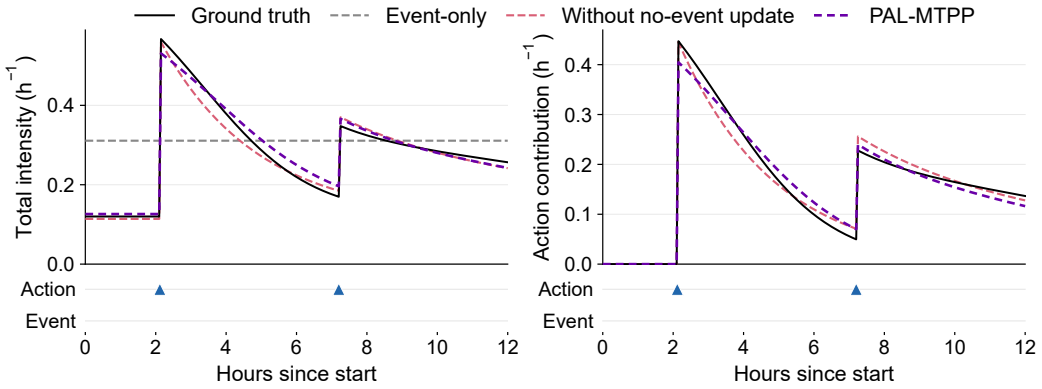
(2) PAL-MTPP without no-event updates, which updates lifecycle states only from observed events;

(3) PAL-MTPP, which uses both event and no-event observations.

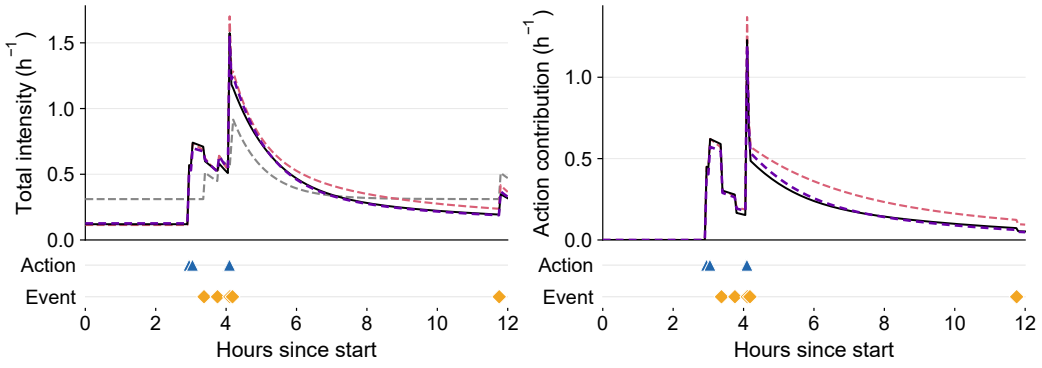
The recovered total intensities are evaluated using intensity RMSE and NLL. Figure 8 further visualizes the recovered intensity trajectories. The figure compares the ground-truth intensity with the intensities estimated by different variants. The results show that incorporating prospective actions improves intensity recovery compared with event-only modeling. Removing no-event updates leads to higher recovery error, indicating that event absence provides additional information for lifecycle inference.

Table 3: Synthetic intensity recovery.

Model	RMSE ↓	NLL ↓
Event-only	0.275	8.814
PAL-MTPP (without no-event updates)	0.044	7.938
PAL-MTPP	**0.030**	**7.929**



(a) No observed events



(b) Overlapping actions

Figure 8: Synthetic intensity recovery. Ground-truth and recovered trajectories under different model variants. Left: total intensity. Right: action contribution.

C Additional Details

C.1 Lifecycle Update Implementation

Section 3.3 defines the factorized approximate filtering mechanism used by PAL-MTPP. Each prospective action enters the pending state when it is first observed, and the model maintains its marginal pending probability $\pi_i(t)$ rather than the full joint lifecycle posterior.

Between events, the pending probabilities are updated using the no-event recursion in Eq. 4. When a target-matching event occurs, PAL-MTPP updates the action marginals according to the possible baseline and action-realization explanations rather than treating the event as a deterministic realization. Explicit cancellation observations remove the corresponding action from the pending state. After each event update, only the marginal pending probabilities are retained, consistent with the factorized approximation.

C.2 Multi-target Construction

Section 4 models multi-target events using a target-set distribution. Here we describe how target sets are constructed for the e-commerce datasets.

For Synerise, purchase records from the same user and timestamp are grouped into one target set. For Retailrocket, items belonging to the same transaction are grouped into one target set.

Cart additions are treated as prospective actions associated with possible future purchase targets, while purchase or transaction records provide realization observations for the corresponding targets. Singleton purchases are retained as target sets with cardinality one and are handled by the same prediction framework.

C.3 Baseline

Section 6.1 introduces the baseline models used in the main comparison. Here we describe how these models are adapted to the prediction tasks considered in this work.

NHP, THP, and S2P2 are used as event-history encoders. Within each domain, the compared models use the same candidate sets and prediction heads, so the main difference lies in how historical information is represented rather than the output space.

Action-Feature and Action-Transformer use the same prospective-action records. Action-Feature aggregates action records into fixed features, while Action-Transformer encodes individual action records using attention. Both models use the same prediction setting as PAL-MTPP within each domain.

For lifecycle analysis, comparisons are performed by changing the lifecycle update mechanism while keeping the input data and model architecture unchanged. For interaction analysis, the proposed pairwise refinement is compared with additive target corrections under the same evaluation protocol.

C.4 Dataset Sources and Evaluation Protocol

We use three publicly available datasets: LBS-IntentBench, Synerise, and Retailrocket.

The evaluation protocol differs across datasets. LBS-IntentBench uses five-fold out-of-fold evaluation, while Synerise and Retailrocket use chronological splits. For the Synerise and Retailrocket e-commerce datasets, validation data are used for checkpoint selection. Because the splits are defined by time, the same user may appear in multiple partitions. Table 4 summarizes the dataset sizes under these evaluation protocols.

For result aggregation, LBS-IntentBench uses three random seeds (101, 202, and 303). For each seed, predictions from the five held-out folds are concatenated at the user level before computing the evaluation metrics, and the final results are averaged across the three seeds. Synerise and Retailrocket use three random seeds (11, 22, and 33) with fixed temporal splits. Metrics are computed independently for each seed and then averaged across seeds.

Table 4: Dataset statistics under the evaluation protocols.

Dataset / evaluation split	Users	Risk intervals	Scored events / baskets
LBS-IntentBench: five-fold OOF total	781	24,268	23,489
Synerise: train	55,248	193,395	18,728
Synerise: validation	8,834	32,028	2,664
Synerise: test	8,732	31,781	2,592
Retailrocket: train	23,349	188,912	10,278
Retailrocket: validation	8,736	57,052	3,215
Retailrocket: test	8,243	52,180	2,854

C.5 Training Details

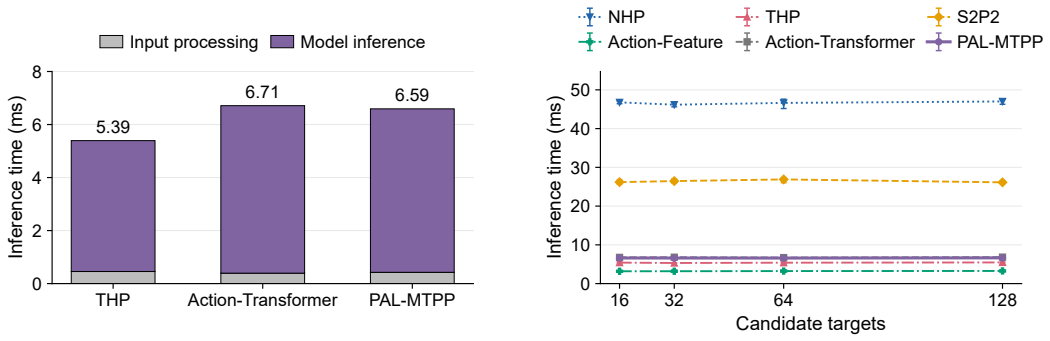
Section 6.1 summarizes the experimental protocol. We provide additional optimization details for reproducibility. All models are optimized using Adam with learning rate 10^{-3}. Mobility experiments use five user folds and three random seeds. Commerce experiments use three random seeds. Model checkpoints are selected according to validation Joint NLL. The main training configurations are summarized in Table 5.

Table 5: Training configurations used in the experiments.

Setting	Mobility	Synerise	Retailrocket
Optimizer	Adam	Adam	Adam
Learning rate	10^{-3}	10^{-3}	10^{-3}
Batch size	Full batch	512	512
Random seeds	3	3	3

D Inference Time Analysis

We evaluate the inference time of PAL-MTPP compared with baseline models. Figure 9 reports inference time under different candidate target sizes. The left panel decomposes runtime into input processing and model inference components. The right panel shows how inference time changes with the number of candidate targets. Within the evaluated range, PAL-MTPP introduces limited additional overhead while maintaining comparable inference time to action-based baselines. All measurements are performed on controlled in-memory Synerise workloads with 32 history records and batch size one using an RTX 5060 Ti GPU. Each measurement averages 30 repetitions across three checkpoints.



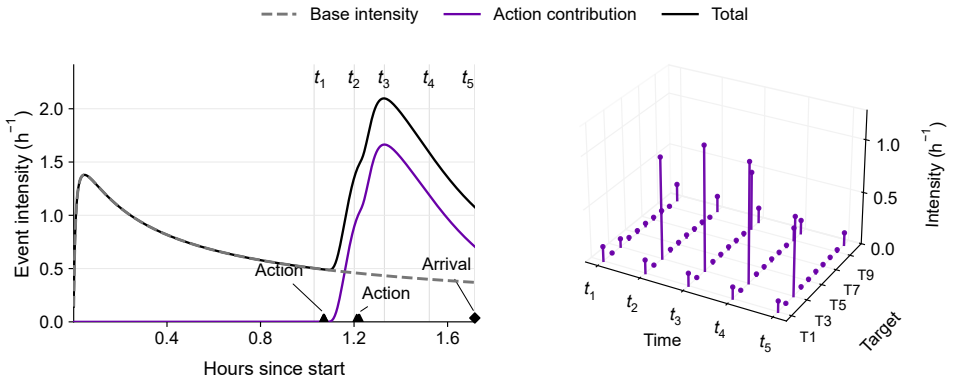
(a) Runtime breakdown

(b) Cost across candidate sizes

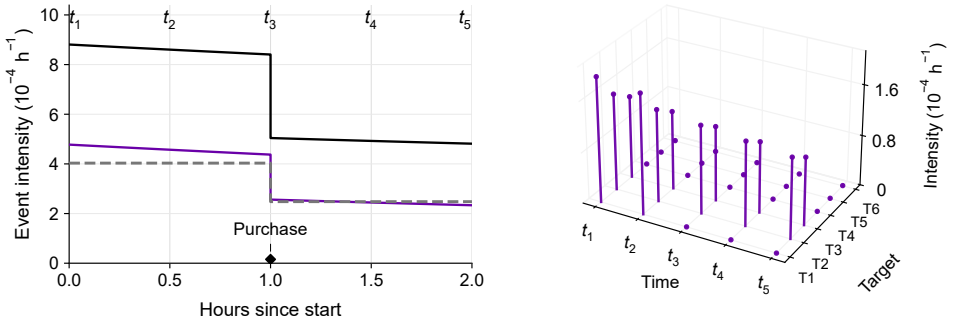
Figure 9: Inference time analysis of PAL-MTPP and baseline models. (a) Runtime breakdown. (b) Inference time under different candidate target sizes.

E VISUALIZATIONS

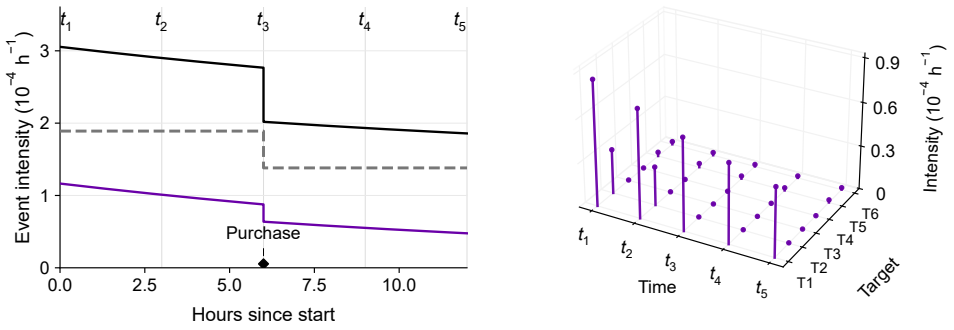
This section visualizes the intensity dynamics produced by PAL-MTPP on the three real-world datasets. Figure 10 presents examples from LBS-IntentBench, Synerise, and Retailrocket. For each example, the left panel decomposes the total event intensity into the baseline intensity and prospective-action contribution over time, while the right panel shows the corresponding target-level intensities at selected time points. Across the three datasets, the prospective-action component adds a time-varying contribution to the baseline intensity. In LBS-IntentBench, newly observed actions increase the intensity of their associated targets. In the two e-commerce examples, purchase observations reduce the remaining action contribution, reflecting lifecycle updates after realization. The target-level views further show that these changes are concentrated on a subset of targets rather than applied uniformly across the target space.



(a) LBS-IntentBench



(b) Synerise



(c) Retailrocket

Figure 10: Qualitative visualization of intensity dynamics learned by PAL-MTPP. The examples show how baseline intensity and prospective-action contributions evolve over time in mobility and e-commerce datasets.