Deliberate-When-Needed: Flow-Reasoner for Neuro-Symbolic Continuous Thought

Wenjie Shen* **Boyang Li*** **Chao Yang** **Shuang Li†**
The Chinese University of Hong Kong, Shenzhen
School of Data Science

Abstract

We present **Flow-Reasoner**, a Deliberate-When-Needed neuro-symbolic model that integrates continuous latent cognition with selective symbolic reasoning. The mental module is a latent state vector evolving smoothly under a first-order ordinary differential equation (ODE), capturing continuous thought that drifts and decays between interventions. The action module is a temporal point process whose intensities are modulated by symbolic rules. Crucially, reasoning is not constant: it is triggered only at irregular instants—when an observed action arrives or when a latent state crosses a threshold—at which point a bounded differentiable forward-chaining procedure updates beliefs and adjusts event likelihoods. Between these triggers, cognition evolves autonomously under the ODE without symbolic intervention. This design yields a model that (i) unifies continuous-time dynamics with selective logical reasoning, (ii) predicts both the type and timing of future actions, and (iii) produces concise rule traces that explain predictions. Empirical studies on synthetic benchmarks and real-world behavioral datasets demonstrate that Flow-Reasoner consistently outperforms strong temporal point process baselines, while providing interpretable, cognitively inspired explanations of decision dynamics. The code is publicly available at `https://github.com/shennnnwj/flow-reasoner`.

*Equal Contribution. †Corresponding Author. Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

1 INTRODUCTION

Human behavior unfolds as a continuous stream in which *mental states* and *actions* co-evolve (Gallagher, 2020). Latent mental states—such as urgency or attention—change gradually over time, while observable actions occur at irregular moments. For example, frustration may build before an abrupt outburst, or a decision may slowly form before surfacing as a discrete action (Binz et al., 2025). Capturing such patterns requires models that represent both the continuous evolution of latent mental states and the discrete events through which they shape behavior.

Existing approaches to modeling human behavior capture only fragments of this picture. Neural temporal point process models achieve strong predictive performance for event types and timing, but rely on largely black-box hidden states that are typically updated only at observed event times (Du et al., 2016; Mei and Eisner, 2017; Zuo et al., 2020). Neural ODE-based models naturally represent continuous latent trajectories, yet lack explicit mechanisms for explaining how latent-state dynamics influence future events (Chen et al., 2018, 2020; Han et al., 2024). Neuro-symbolic models *introduce structured reasoning that can make such influences more explicit*, but are often formulated in discrete time or assume reasoning is continuously active rather than episodic (Serafini and Garcez, 2016; Yang et al., 2024). As a result, prior work does not jointly capture continuously evolving latent mental states, irregular action events, and sparse, event- or state-triggered reasoning within a unified framework. More broadly, while much of the literature focuses on how to incorporate reasoning, it pays far less attention to *when reasoning should occur*.

We address this gap with **Flow-Reasoner**, a deliberate-when-needed neuro-symbolic framework for continuous-time action prediction. The key idea is to decouple continuous latent-state evolution from discrete symbolic reasoning while explicitly modeling how the former triggers the latter and shapes future actions. Instead of performing symbolic inference

throughout the entire trajectory, Flow-Reasoner maintains continuously evolving latent mental states and invokes reasoning at discrete instants—such as when a new action is observed or when the latent state crosses a trigger threshold. This reasoning step updates the model's internal state and directly influences both the type and timing of the next action event. In this view, *continuous thought* refers to the evolution of latent mental states, while symbolic reasoning occurs sparsely but plays a causal role in shaping irregular action events.

A key novelty of Flow-Reasoner is modeling both when and how to reason. At each reasoning step, the model performs a bounded differentiable multi-step forward-chaining update over softly matched rule templates. The resulting rule activations propagate across time, modulating latent-state evolution and influencing future event likelihoods. This event- and state-triggered mechanism treats reasoning as an intermittent update to continuous dynamics, rather than a static prior or always-on constraint. The result is a unified integration of continuous-time latent dynamics, structured rule-based reasoning, and asynchronous event prediction.

This design yields three benefits. First, it unifies latent dynamics, event modeling, and multi-step reasoning in a single end-to-end framework. Second, it enables sparse, state-aware reasoning concentrated at behaviorally meaningful moments. Third, it preserves interpretability via rule-level traces while remaining computationally efficient. Rules act as learned structural biases whose matching and influence are refined through training, rather than as fixed symbolic programs.

Empirically, Flow-Reasoner improves both event-time and event-type prediction on synthetic and real-world behavioral datasets. Gains arise from the interaction between continuous dynamics, triggered reasoning, and multi-step inference. The model also provides concise rule-level explanations, and supplementary experiments show that triggered reasoning substantially reduces the cost of always-on reasoning while remaining robust to incomplete rule sets.

This paper makes three contributions:

(*i*) We introduce Flow-Reasoner, a continuous-time neuro-symbolic framework that couples latent mental-state evolution with event prediction and triggers reasoning only at event or state-dependent instants.

(*ii*) We propose an end-to-end model integrating continuous dynamics with differentiable multi-step rule reasoning via bounded forward chaining over soft rule templates.

(*iii*) We provide extensive empirical evidence on synthetic and real-world behavioral datasets showing improved predictive accuracy, concise rule-level interpretability, substantial computational savings over always-on reasoning, and robustness to imperfect rule sets.

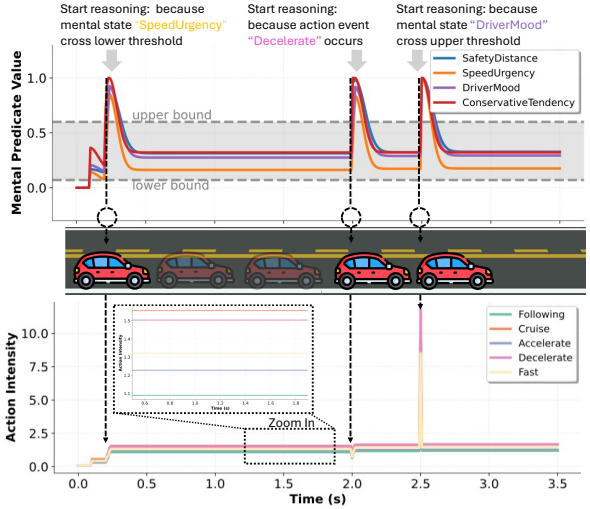

Figure 1: Latent mental states (e.g., speed urgency, driver mood) evolve continuously and trigger reasoning at threshold crossings or driving events (e.g., decelerate). Reasoning updates rule activations, producing sharp changes in mental states and action intensities (e.g., accelerate, fast), which persist until the next trigger.

2 RELATED WORK

Human behavior modeling with latent internal states A central problem in human behavior modeling is to explain observable actions in terms of latent internal variables such as intention, urgency, attention, or task state. Prior work has introduced latent-variable models for human activity generation and intention inference, including variational and sequential models that infer hidden behavioral factors from action trajectories (Mehrasa et al., 2019; Jeong et al., 2021; Zolotas and Demiris, 2022). A related line of work seeks more interpretable accounts of behavior through symbolic or structured models, including logic-based models for human action explanation (Cao et al., 2023) and recent efforts to discover symbolic cognitive models directly from behavioral data (Castro et al., 2025). These works highlight the importance of latent internal structure, but they typically operate in discrete time, assume stepwise latent updates, or do not model irregular action timing in continuous time.

Continuous-time models for irregular behavioral events When human behavior is observed as asynchronously timed actions, temporal point processes (TPPs) provide a natural modeling framework (Hawkes, 1971; Daley and Vere-Jones, 2008;

Rasmussen, 2018). Neural TPPs such as RMTPP, Neural Hawkes, SAHP, THP, and related attention-based variants achieve strong predictive performance for event type and timing by learning expressive continuous-time representations from event histories (Du et al., 2016; Mei and Eisner, 2017; Zhang et al., 2020; Zuo et al., 2020; Yang et al., 2021). In parallel, Neural ODEs and related continuous-time latent-variable models provide flexible tools for modeling smoothly evolving hidden states under irregular observations (Chen et al., 2018; Rubanova et al., 2019; De Brouwer et al., 2019; Jia and Benson, 2019; Kidger et al., 2020; Chen et al., 2020; Han et al., 2024). These approaches are well suited to continuous-time prediction, but their latent dynamics are typically learned as black-box representations and are not coupled to explicit symbolic reasoning about human behavior.

Logic-informed temporal reasoning for behavioral sequences Another line of work incorporates symbolic structure into temporal event modeling. Temporal Logic Point Processes, interpretable logic-rule learning for TPPs, Weighted Clock Logic Point Processes, knowledge-informed TPPs, and latent causal rule models introduce logic into event intensities or temporal dependencies to improve interpretability and inject domain structure (Li et al., 2020, 2021; Yan et al., 2023; Zhang et al., 2021; Kuang et al., 2024). Neural Datalog Through Time further shows how logical specifications can be compiled into continuous-time temporal models with time-varying facts (Mei et al., 2020). More recently, Neuro-Symbolic Temporal Point Processes learns compact temporal rules and their weights end-to-end within a TPP framework (Yang et al., 2024), while Evolving Minds performs logic-informed inference over temporal action patterns to recover latent mental events from observed behavior (Yang et al., 2025). These works are closely related to ours, but they generally attach logic directly to event prediction or latent fact inference. They do not maintain a separately evolving latent mental-state process together with sparse multi-step reasoning updates that shape both subsequent latent evolution and future actions.

3 BACKGROUND

We briefly review the mathematical components underlying our framework: first-order ODEs for latent mental states, temporal point processes (TPPs) for observable actions, and rule-based reasoning via forward chaining.

3.1 Latent Dynamics via ODEs

We represent latent mental states as a continuous vector $M(t) \in \mathbb{R}^d$. Its evolution is governed by a first-order ordinary differential equation (ODE):

$$\frac{dM}{dt} = f(M, t; \theta), \tag{1}$$

where f is a parameterized drift function. Between reasoning instants, $M(t)$ evolves smoothly under this dynamics, capturing gradual accumulation and decay of mental activations.

3.2 Temporal Point Processes (TPPs)

Observable actions are modeled as marked temporal point processes $\mathcal{E} = \{(t_i, a_i)\}_{i=1}^{N}$, with $t_i \in \mathbb{R}^+$ the event time and $a_i \in \mathcal{A}$ the action type. Let $\mathcal{H}_t = \{(t_j, a_j) \mid t_j < t\}$ denote the event history up to time t. The conditional intensity for action type a is

$$\lambda(t, a \mid \mathcal{H}_t) = \lim_{\Delta t \to 0} \frac{\Pr[\text{event} \in [t, t+\Delta t) \mid \mathcal{H}_t]}{\Delta t}, \tag{2}$$

with total intensity defined as $\lambda(t \mid \mathcal{H}_t) = \sum_{a \in \mathcal{A}} \lambda(t, a \mid \mathcal{H}_t)$. A generative view is given by sampling the next event time from the survival function,

$$\Pr[T > \tau \mid \mathcal{H}_{t_i}] = \exp\Big(- \int_{t_i}^{\tau} \lambda(s \mid \mathcal{H}_s)\, ds \Big), \tag{3}$$

and then sampling the action type conditional on that time,

$$\Pr(a = k \mid t, \mathcal{H}_t) = \frac{\lambda(t, k \mid \mathcal{H}_t)}{\lambda(t \mid \mathcal{H}_t)}. \tag{4}$$

This framework naturally captures irregularly spaced human actions in continuous time.

3.3 Rules and Forward Chaining

We represent symbolic knowledge as a set of rules defined over predicates. A predicate is a Boolean function $P : X \to \{0, 1\}$, where X is the space of trajectories. Each rule $f \in F$ is expressed as a definite Horn clause:

$$f : \; P_0(x) \leftarrow P_1(x) \wedge P_2(x) \wedge \cdots \wedge P_h(x), \quad x \in X,$$

where P_0 is the head and $P_1, \ldots, P_h$ form the body.

Forward chaining is a rule-based inference procedure that starts from known predicates and iteratively applies rules whose bodies are satisfied to derive new predicates. The newly inferred predicates are then fed back into the process, enabling multi-step reasoning until no further updates occur (i.e., closure is reached).

Intuitively, this process resembles step-by-step reasoning. For example, observing that "the road is wet" and "it is raining" may lead to the inference "driving is risky," which can further support "slow down." Each conclusion becomes a new piece of evidence for subsequent reasoning, forming a chain of inferences rather than a single-step decision.

These elements provide the foundation for Flow-Reasoner: latent mental states evolve continuously under ODE dynamics, action events are governed by a TPP, and forward-chaining rules link the two through intermittent reasoning updates.

4 METHOD

Flow-Reasoner combines two coupled components: (i) a marked temporal point process for irregular *human actions*, and (ii) a continuous latent process for *mental states* evolving over time. The first captures discrete actions and their timing, while the second represents gradually changing internal conditions that are not directly observed. Figure 1 illustrates this interplay in a car-following scenario, where irregular actions are interleaved with continuously evolving latent mental states.

To connect these components, we introduce structured reasoning as the mechanism linking latent mental states and observations to future actions. Reasoning is performed only at discrete *reasoning instants*, triggered either by new observations (e.g., an action arrival) or when latent mental states cross predefined thresholds indicating a need for update. Between reasoning instants, no further symbolic inference is performed; instead, latent mental states evolve continuously according to the ODE dynamics, and their influence on actions remains fixed until the next reasoning update.

Figure 2 provides an overview of the Flow-Reasoner architecture. We next describe each component and their interactions in detail.

4.1 Actions and Latent Mental States

Rule system We consider a rule set $\mathcal{R}$, partitioned into three *rule families*: (i) $A \to M$ (actions influence mental states), (ii) $M \to A$ (mental states influence actions), and (iii) $M \to M$ (interactions among mental states). These families capture complementary dependencies: actions provide evidence for updating latent mental states, mental states guide future actions, and mental states refine each other through internal interactions. Together, they define a unified rule system for modeling the bidirectional coupling between actions and latent mental dynamics in Flow-Reasoner.

Each rule $f \in \mathcal{R}$ has an activation score $g_f(\rho) \in [0,1]$, computed by Flow-Reasoner at reasoning instant ρ via differentiable forward chaining (see Sec. 4.2).

Latent mental dynamics (ODE). Between two consecutive reasoning instants $\rho < \rho'$, each mental state follows:

$$\frac{dM_j}{dt} = -\alpha_j M_j + (1 - M_j)\, S_j^{\langle\rho\rangle}, \qquad \alpha_j > 0, \tag{5}$$

where α_j is a learnable decay rate and

$$S_j^{\langle\rho\rangle} = \sum_{f \in \mathcal{R}_{A\to M,j}} \gamma_{j,f}\, g_f(\rho).$$

Here, $S_j^{\langle\rho\rangle}$ is the total logic-induced excitation for mental dimension j, computed once at ρ and held constant until the next reasoning instant. Nonnegative weights $\gamma_{j,f} \geq 0$ encode the strength of rule f's influence on mental dimension j.

Since $S_j^{\langle\rho\rangle}$ is constant on $(\rho, \rho']$, Eq. (5) admits the solution:

$$M_j(t) = M_j(\rho)\, e^{-(\alpha_j + S_j^{\langle\rho\rangle})(t-\rho)} + \frac{S_j^{\langle\rho\rangle}}{\alpha_j + S_j^{\langle\rho\rangle}} \Big(1 - e^{-(\alpha_j + S_j^{\langle\rho\rangle})(t-\rho)}\Big).$$

This ensures bounded trajectories and monotonic relaxation toward a logic-dependent equilibrium.

Observed action events (TPP) We model each action type $k \in \{1, \ldots, K\}$ with a conditional intensity:

$$\lambda_k(t \mid \mathcal{H}_t) = b_k + B_k^{\langle\rho\rangle}, \qquad b_k \geq 0, \tag{6}$$

where b_k is a learnable baseline rate.

The logic-induced boost is defined by:

$$B_k^{\langle\rho\rangle} = \sum_{f \in \mathcal{R}_{M\to A,k}} \gamma_{k,f}\, g_f(\rho),$$

where nonnegative weights $\gamma_{k,f} \geq 0$ encode the strength of rule f's influence on action intensity of dimension k.

Flow-Reasoner evaluates all rules $\mathcal{R}$ only at reasoning instants. The resulting activations $g_f(\rho)$ act as piecewise-constant covariates, which modulate both (i) mental evolution via $S_j^{\langle\rho\rangle}$ and (ii) event intensities via $B_k^{\langle\rho\rangle}$. Between instants, the system evolves continuously with frozen logic context, yielding a hybrid dynamical system with piecewise-constant survival terms.

Predicate Values (actions and mental states) To enable symbolic reasoning over both observed actions and latent mental states, we represent them *using a unified notion of predicate values*, while preserving their different temporal nature.

Mental predicates correspond to latent dimensions of $M(t)$. For a mental predicate $K \equiv M_j$, its value is given directly by

$$\textsc{val}(K, t) = M_j(t).$$

Action predicates represent discrete events and are converted into continuous signals via a decaying memory of the most recent occurrence. Let $\tau_K(t)$ denote the most recent time action K occurred (or $-\infty$ if it has never occurred). Its predicate value is

$$\textsc{val}(K, t) = \begin{cases} \exp\big(-\lambda(t - \tau_K(t))\big), & \text{if } \tau_K(t) > -\infty, \\ 0, & \text{otherwise,} \end{cases}$$

where $\lambda > 0$. This construction yields continuous-valued representations of both mental states and actions, enabling them to be used jointly as inputs for multi-step symbolic reasoning.

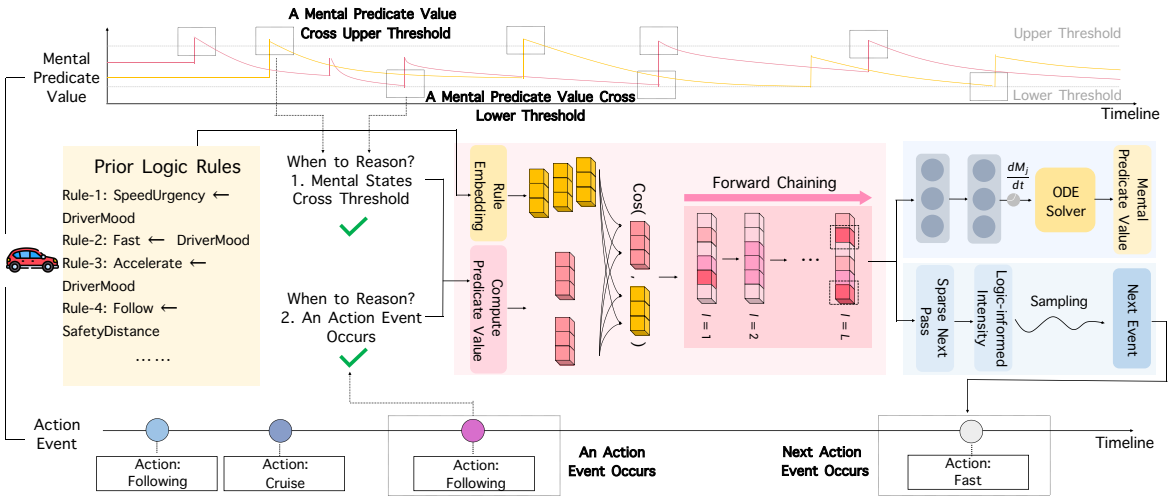

Figure 2: Overview of Flow-Reasoner. Observed actions and latent mental states define predicate values, which are matched to rule templates through learned embeddings. Forward chaining is executed only at reasoning instants, and the resulting rule scores are cached into $S^{\langle\rho\rangle}$ for the latent ODE and $B^{\langle\rho\rangle}$ for action intensities until the next reasoning instant.

4.2 Neural-Symbolic Forward Chaining at Reasoning Instants

At each reasoning instant, Flow-Reasoner performs *neural-symbolic forward chaining* over a set of learnable logic rules. Unlike fixed symbolic programs, the rule structure is parameterized and can be learned or refined during training, allowing the model to adapt its reasoning patterns from data while preserving an interpretable rule form.

Predicate dictionary and representations We maintain a predicate dictionary $\mathcal{K} = \mathcal{K}_A \cup \mathcal{K}_M$, where $\mathcal{K}_A$ contains action predicates and $\mathcal{K}_M$ contains mental predicates. This dictionary indexes the same predicates whose time-varying values $\textsc{val}(K, t)$ were defined in Sec. 4.1. Thus, each predicate has two complementary representations: (i) a dynamic value $\textsc{val}(K, t)$ capturing its current activation over time, and (ii) a fixed (anchored) embedding used only for semantic identification.

Soft grounding of rule slots Each rule f is parameterized by slot embeddings $\Theta_f = [\theta_1, \ldots, \theta_h]$, which are grounded into the predicate dictionary $\mathcal{K}$.

Instead of hard symbolic binding, each slot induces a soft alignment over predicates based on cosine similarity:

$$\pi_j(K) = \mathrm{softmax}_\tau\big(\{\cos(\theta_j, K')\}_{K' \in \mathcal{K}}\big).$$

This defines a differentiable distribution over predicate identities for each slot. We write:

$$K_j^\star \sim \pi_j(\cdot),$$

to denote that slot j is softly associated with predicates in $\mathcal{K}$.

The corresponding predicate value used in forward chaining is computed as a soft retrieval over predicate states:

$$\textsc{val}(K_j^\star, t) = \sum_{K \in \mathcal{K}} \pi_j(K)\, \textsc{val}(K, t).$$

This formulation preserves discrete predicate structure while enabling differentiable grounding of rule slots, allowing smooth matching between rule templates and predicate symbols during learning and reasoning.

Differentiable forward chaining We introduce a differentiable relaxation of classical forward chaining, where discrete rule firing is replaced by continuous activations computed at each reasoning instant.

(I) One-hop inference via rule evaluation A single forward-chaining hop computes how strongly each rule fires and how its head predicates are updated.

AND: rule body evaluation For each rule f with h slots, we compute its body satisfaction using a soft-min aggregation:

$$g_f^{\mathrm{AND}}(t) = \mathrm{softmin}_\tau\big(\{\textsc{val}(K_j^\star, t)\}_{j=1}^h\big),$$

where

$$\mathrm{softmin}_\tau(\mathbf{x}) = -\frac{1}{\tau} \log \frac{1}{|\mathbf{x}|} \sum_i \exp(-x_i/\tau).$$

This measures whether all predicates in the rule body are simultaneously satisfied.

OR: rule aggregation If a head predicate Q is supported by multiple rules $f_1, \ldots, f_R$, their contributions are aggregated via a soft-OR:

$$g_Q(t) = \frac{1}{\beta} \log \sum_{r=1}^{R} \exp\big(\beta\, g_{f_r}^{\mathrm{AND}}(t)\big).$$

One-hop update. The aggregated score $g_Q(t)$ is used to update the predicate value:

$$\textsc{val}^{(1)}(Q,t) \leftarrow \max\big(\textsc{val}^{(0)}(Q,t),\, g_Q(t)\big).$$

This defines a single *forward-chaining hop*, where rules are applied once to propagate evidence and update predicate values.

(II) Multi-hop forward chaining at a reasoning instant At each reasoning instant ρ, Flow-Reasoner performs multi-hop inference by iteratively applying the one-hop update for L steps, where each hop refines predicate values based on the previous one.

We initialize $\textsc{val}^{(0)}(K,\rho)$ for all predicates K, using last-hit kernels for action predicates and the left-limit state $M(\rho^-)$ for mental predicates.

After L iterations, we obtain refined activations $\{g^{(\ell)}(\rho)\}_{\ell=1}^{L}$ and final predicate values $\textsc{val}^{(L)}(\cdot,\rho)$ (Alg. 1), yielding a bounded fixed-point approximation of forward chaining.

These outputs update both action-level intensity contributions and mental-state driving signals, producing the post-reasoning state $M(\rho^+)$. This state serves as the initial condition for ODE evolution until the next reasoning instant.

4.3 When and how Flow-Reasoner reasons: intermittent reasoning instants

In Flow-Reasoner, mental states evolve continuously via an ODE, while symbolic reasoning is executed only at discrete *reasoning instants*. This separates smooth temporal evolution from discrete logical inference.

We define reasoning instants $\mathcal{T}_{\mathrm{R}} = \{\rho_0 < \rho_1 < \cdots\}$, at which the multi-hop forward-chaining procedure (Sec. 4.2) updates predicate values.

Triggering mechanisms Reasoning is triggered by:

(i) **Events:** each action time $t_i \in \mathcal{T}_{\mathrm{R}}$ introduces new evidence.

(ii) **State changes:** a new instant is triggered when any $M_j(t)$ crosses the hysteresis band $[\underline{h}_j, \overline{h}_j]$, indicating a qualitative regime shift.

Table 1: Intermittent reasoning cycle in Flow-Reasoner.

Phase	Instant	Description
1	ρ	Trigger by event or state change.
2	ρ	Run multi-hop forward chaining to compute $\textsc{val}(\cdot,\rho)$.
3	ρ^+	Update $M(\rho^+)$ and cache reasoning outputs.
4	(ρ,ρ')	Continuous ODE evolution with fixed signals.

Algorithm 1 Multi-hop ForwardChaining(ρ, L)

Require: initial values $\textsc{val}^{(0)}(\cdot,\rho)$, rule set $\mathcal{R}$, slot embeddings $\{\Theta_f\}$, temperatures (τ,β), max hops L, tolerance ε

$\ell \leftarrow 1$, $\Delta \leftarrow +\infty$

while $\ell \le L$ **and** $\Delta > \varepsilon$ **do**

  (1) Rule body evaluation (AND):

  for $f \in \mathcal{R}$ **do**

$$g_f^{(\ell)}(\rho) \leftarrow \mathrm{softmin}_\tau\Big(\{\textsc{val}^{(\ell-1)}(K_j^\star,\rho)\}_{j=1}^{h_f}\Big)$$

  end for

  (2) Head aggregation (OR):

  for each head predicate Q **do**

$$G_Q^{(\ell)}(\rho) \leftarrow \frac{1}{\beta}\log \sum_{f\in\mathcal{F}(Q)} \exp\big(\beta\, g_f^{(\ell)}(\rho)\big)$$

  end for

  (3) One-hop update:

  for each predicate Q **do**

$$\textsc{val}^{(\ell)}(Q,\rho) \leftarrow \max\Big(\textsc{val}^{(\ell-1)}(Q,\rho),\, G_Q^{(\ell)}(\rho)\Big)$$

  end for

  (4) Convergence check:

$$\Delta \leftarrow \max_Q \Big|\textsc{val}^{(\ell)}(Q,\rho) - \textsc{val}^{(\ell-1)}(Q,\rho)\Big|$$

  $\ell \leftarrow \ell + 1$

end while

Return $\textsc{val}^\star(\cdot,\rho)$ and rule scores $g_f(\rho)$

As summarized in Tab. 1, at each $\rho \in \mathcal{T}_{\mathrm{R}}$, Flow-Reasoner runs multi-hop forward chaining to compute $\textsc{val}(\cdot,\rho)$. The resulting outputs update action intensities and mental-state driving signals, yielding the post-reasoning state $M(\rho^+)$. This state initializes the ODE evolution until the next reasoning instant. Between instants, no symbolic reasoning is performed and the system evolves continuously under fixed reasoning outputs.

5 MODEL LEARNING

5.1 Parameter learning

Flow-Reasoner learns parameters across three components: (i) rule structure parameters $\Theta_f = [\theta_1,\ldots,\theta_h]$, $f \in \mathcal{R}$, (ii) rule weights $\{\gamma_{j,f},\gamma_{k,f}\}$, (iii) baseline intensity and decaying rate of ODEs $\{\alpha_j, b_k\}$.

Marked point process likelihood We optimize the standard marked temporal point process objective:

$$\mathcal{L} = -\sum_{i=1}^{N} \log \lambda_{a_i}(t_i \mid \mathcal{H}_{t_i}) + \int_0^T \sum_{k=1}^{|\mathcal{K}_A|} \lambda_k(\tau \mid \mathcal{H}_\tau)\, d\tau.$$

Between reasoning instants $\rho \in \mathcal{T}_{\mathrm{R}}$, intensities are constant: $\lambda_k(\tau) = b_k + B_k^{\langle\rho\rangle}$ for $\tau \in (\rho, \rho']$, which yields:

$$\int_0^T \lambda_k(\tau)\, d\tau = \sum_{(\rho,\rho']} (b_k + B_k^{\langle\rho\rangle})(\rho' - \rho).$$

This reduces training to a covariate-driven Poisson process with piecewise-constant parameters updated only at reasoning instants.

Stochastic training with intermittent reasoning To estimate parameters we maximize the marked-TPP log-likelihood under the piecewise-constant intensities. Because reasoning is invoked only at discrete instants, training proceeds segment by segment: at each instant we (i) run forward chaining (Alg. 1) to refresh rule activations, (ii) cache inputs for the latent ODE (5) and intensity (6), (iii) propagate the mental state continuously until the next instant, and (iv) accumulate both survival loss and event likelihood. This cycle is repeated over mini-batches of subsequences, with gradients backpropagated end-to-end through both the reasoning and ODE modules.

5.2 Hyperparameter selection

Number of reasoning hops The number of forward-chaining iterations L is a hyperparameter controlling the depth of logical refinement at each reasoning instant. Larger L improves logical completeness but increases computational cost and may introduce optimization instability. In practice, we use a small fixed L for efficiency, and select its value via cross-validation based on predictive performance.

Hysteresis thresholds The state-triggered reasoning thresholds $[\underline{h}_j, \overline{h}_j]$ operate on the bounded mental-state variables, where each $M_j(t) \in [0, 1]$ represents a predicate-like activation. To avoid overly frequent triggering early in training, we first disable state-triggered reasoning and rely only on event-triggered instants. We then gradually introduce hysteresis-based triggering by tuning $[\underline{h}_j, \overline{h}_j]$ on a validation set, balancing reasoning frequency and predictive performance. This allows the model to progressively discover stable internal regime changes.

6 EXPERIMENT

6.1 Experiment Setup

Datasets. We evaluate Flow-Reasoner on one synthetic dataset and two real-world behavioral datasets. The synthetic dataset provides a controlled setting in which the symbolic structure used for data generation is known, while training is performed with a separate candidate rule pool. Specifically, the synthetic dataset contains 2,000 sequences, a fixed time horizon of 15 seconds, and a fixed set of predicates and logic rules; the candidate rule pool used in training is constructed by perturbing the data-generation rules, so the model must learn which rules are useful and how strongly they should influence the dynamics.

For real-world data, we consider two behavior-centric corpora that are plausibly driven by latent mental states: *Hand-Me-That* (Wan et al., 2022), where we use the change-state episodes and obtain 500 sequences with an average action-trajectory length of 30.5, and *Car-Following* (Li et al., 2023), from which we extract 5,000 sequences, each containing on average 3.6 action events and a mean time horizon of 7.20 seconds. In all experiments, only action trajectories are observed during training and mental states are treated as latent variables that serve only as intermediate representations for action prediction; in the synthetic setting, ground-truth data-generation rules are masked. Accordingly, all evaluations are reported in terms of action-type and event-time prediction. For fairness, all baselines are given the same event sequences, but none of them includes a symbolic rule layer.

Baselines We compare against baselines from three categories: neural temporal point process models, logic-based models, and generative models. The neural TPP baselines include RMTPP (Du et al., 2016), THP (Zuo et al., 2020), PromptTPP (Xue et al., 2023), and HYPRO (Xue et al., 2022); the logic-based baselines include TELLER (Li et al., 2021), CLNN (Yan et al., 2023), and STLR (Cao et al., 2023); and the generative baselines include AVAE (Mehrasa et al., 2019), GNTPP (Lin et al., 2022), VEPP (Pan et al., 2020), and STVAE (Wang et al., 2023). For PromptTPP and HYPRO, we use AttNHP (Yang et al., 2021) as the base model. For GNTPP, we adopt the revised attentive history encoder and VAE probabilistic decoder.

Comparison Metrics. We follow the common next-event prediction setting in TPPs (Du et al., 2016; Zuo et al., 2020; Yang et al., 2023) and predict the next event from history. In particular, we evaluate prediction along two complementary dimensions: *which action happens next* and *when it happens*. For event-type prediction, we report Top-3 Error Rate (ER%), i.e., the error form of Top-3 accuracy; for event-time prediction, we report Mean Absolute Error (MAE). We also consider autoregressive prediction of multiple future events.

6.2 Experiment on Synthetic Dataset

As shown in Table 2, Flow-Reasoner achieves the best Top-3 ER% on the synthetic dataset while maintain-

Table 2: Performance comparison of different models on various datasets (Top-3).

Category	Model	Syn Data-1		Car-Following		Hand-Me-That	
		ER% ↓	MAE ↓	ER% ↓	MAE ↓	ER% ↓	MAE ↓
Neural TPP	RMTPP	16.55 ± 1.33	2.69 ± 0.09	15.83 ± 1.28	2.64 ± 0.08	45.74 ± 2.17	2.13 ± 0.10
	THP	14.20 ± 1.16	2.33 ± 0.08	13.48 ± 1.57	2.31 ± 0.08	43.88 ± 1.89	2.08 ± 0.12
	PromptTPP	14.17 ± 0.75	2.24 ± 0.07	12.45 ± 0.85	2.11 ± 0.05	42.83 ± 2.09	1.68 ± 0.08
	HYPRO	13.04 ± 1.40	2.17 ± 0.09	12.92 ± 1.66	2.03 ± 0.10	41.06 ± 2.24	1.70 ± 0.08
Rule-based Model	TELLER	17.94 ± 1.66	3.48 ± 0.17	17.04 ± 2.12	3.41 ± 0.22	44.30 ± 2.94	1.86 ± 0.14
	CLNN	18.06 ± 1.29	3.30 ± 0.14	17.11 ± 2.12	3.25 ± 0.18	43.52 ± 1.97	1.82 ± 0.07
	STLR	12.96 ± 1.53	2.81 ± 0.15	12.76 ± 1.89	2.74 ± 0.12	44.25 ± 1.84	1.80 ± 0.09
Generative Model	AVAE	16.37 ± 0.72	3.09 ± 0.08	15.83 ± 1.46	2.95 ± 0.13	46.13 ± 1.79	2.10 ± 0.12
	GNTPP	20.97 ± 1.61	3.92 ± 0.19	20.20 ± 1.59	3.89 ± 0.06	50.36 ± 3.17	2.69 ± 0.12
	VEPP	21.33 ± 1.39	3.85 ± 0.05	20.25 ± 2.14	3.78 ± 0.16	49.74 ± 2.50	2.51 ± 0.11
	STVAE	18.11 ± 1.45	3.26 ± 0.08	17.24 ± 1.68	3.18 ± 0.17	46.12 ± 2.58	2.17 ± 0.09
–	**Ours***	$\mathbf{12.47 \pm 1.49}$	$\mathbf{2.17 \pm 0.04}$	$\mathbf{11.84 \pm 1.01}$	$\mathbf{2.02 \pm 0.07}$	$\mathbf{38.79 \pm 2.19}$	$\mathbf{1.22 \pm 0.11}$

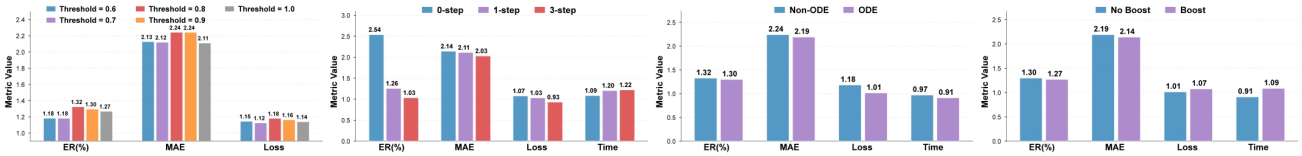

Figure 3: Ablation study on Synthetic Dataset (ER and Loss are divided by 10, and Time is divided by 100 for visualization, so that all metrics remain on a similar scale.)

Table 3: Candidate rule families for the Car-Following dataset, linking action predicates and mental predicates.

A→M Rules
Rule-1: `Accelerate` → `Driver Mood`
Rule-2: `Decelerate` ∧ `Cruise` → `Safety Distance`
Rule-3: `Decelerate` ∧ `Follow` → `Conservative Tendency`
Rule-4: `Accelerate` ∧ `Fast` → `Speed Urgency`
M→M Rules
Rule-1: `Driver Mood` → `Speed Urgency`
Rule-2: `Conservative Tendency` → `Safety Distance`
M→A Rules
Rule-1: `Driver Mood` → `Accelerate`
Rule-2: `Driver Mood` → `Fast`
Rule-3: `Safety Distance` → `Follow`
Rule-4: `Conservative Tendency` → `Decelerate`
Rule-5: `Safety Distance` ∧ `Speed Urgency` → `Cruise`

ing MAE comparable to the strongest neural baseline. This indicates that explicit rules together with selectively triggered reasoning improve *which action happens next*, while the continuous latent ODE dynamics preserve accurate modeling of *when it happens*. In this controlled setting, the simultaneous gains in action-type and event-time prediction support the effectiveness of coupling continuous latent dynamics with deliberate-when-needed symbolic reasoning.

6.3 Experiment on Real-World Datasets

For real-world datasets, we first define a small set of interpretable latent dimensions to organize the continuous mental state. Candidate symbolic rules are then proposed from observed action patterns together with brief domain descriptions, and are treated as weak symbolic candidates rather than fixed oracle logic. Their actual influence is fully determined by learned rule weights during training, so ineffective rules can be down-weighted or effectively removed. As an example, in the Car-Following dataset we identify five action predicates and introduce four latent mental-state predicates as shown in Table 3. Figure 1 illustrates how these latent dimensions evolve together with observed actions, while Table 3 summarizes the three rule families (A→M, M→M, and M→A) used to connect action states and latent mental states.

Table 4: Ablation study on the Synthetic dataset. We vary four modules: (i) **ODE Dynamic**: *Non-ODE* (simple time variation with learnable parameters) vs. *ODE* (closed-form ODE dynamics, our default). (ii) **Neuro-Symbolic Reasoning**: *0-step* (no forward chaining), *1-step*, and *3-step* forward chaining (our default). (iii) **Logic Boost**: *No Boost* (disable logic-strength enhancement) vs. *Boost* (enable logic-strength enhancement, our default). (iv) **Event Thresholding** (upper trigger threshold): *Low*, *Mid* (our default), and *High*. We report ER% (lower is better), MAE (lower is better), training loss, and average training time per epoch.

Ablation Settings				Metrics			
ODE Dynamic	**Neuro-Symbolic Reasoning**	**Logic Boost**	**Event Thresholding**	**ER% ↓**	**MAE ↓**	**Loss ↓**	**Time ↓**
Non-ODE	0-step	No Boost	Mid	13.24	2.24	11.80	97.05
ODE	0-step	No Boost	Mid	12.96	2.19	10.13	90.97
ODE	0-step	Boost	Mid	12.68	2.14	10.75	108.54
ODE	1-step	Boost	Mid	12.56	2.11	10.30	120.40
ODE	3-step	Boost	Mid	10.32	2.03	9.29	122.03
Non-ODE	0-step	No Boost	Low	11.83	2.13	11.46	105.43
Non-ODE	0-step	No Boost	High	12.69	2.11	11.40	105.79

The results in Table 2 show that Flow-Reasoner consistently improves both action-type prediction and event-time prediction over strong baselines on the real-world datasets. The gains are especially pronounced on Hand-Me-That, suggesting that the combination of continuous latent mental states and selective symbolic reasoning is particularly helpful when behavioral sequences are longer and latent cognitive factors are more complex. The consistent gains on Car-Following further indicate that the framework generalizes beyond the controlled synthetic setting to real behavioral data. Taken together, these results show that the proposed coupling of continuous latent dynamics and selectively triggered symbolic reasoning generalizes beyond the synthetic setting.

6.4 Ablation Study

To quantify the contribution of each design choice, we perform ablation experiments on the synthetic dataset. Our default configuration corresponds to *ODE + 3-step + Boost + Mid*. Table 4 reports the predictive performance and training cost of representative configurations, while Figure 3 provides a visual comparison of the same ablation trends.

The ablation results show that all four design choices contribute to the final performance. Replacing the Non-ODE variant with ODE dynamics improves both predictive accuracy and training efficiency; increasing the reasoning depth consistently improves performance, with 3-step reasoning performing best under the full configuration; Logic Boost further reduces ER% and loss in the multi-step reasoning setting; and the medium threshold (Mid) provides a better trade-off between prediction quality and triggering frequency. Overall, the best configuration, *ODE + 3-step + Boost + Mid*, suggests a clear synergy among continuous latent dynamics, multi-step symbolic reasoning, and moderate triggering.

7 CONCLUSION

We present Flow-Reasoner, a differentiable framework for event prediction that couples continuous-time latent mental states with intermittent symbolic reasoning. Mental states evolve continuously via a neural ODE, while bounded multi-hop forward chaining is triggered only at discrete reasoning instants by event arrivals or internal state changes. Across synthetic and real-world behavioral datasets, Flow-Reasoner consistently improves predictive performance. Its key novelty lies in coupling continuously evolving latent dynamics with event/state-triggered symbolic reasoning, rather than applying symbolic reasoning continuously or updating latent dynamics only at event times.

Acknowledgements

This work was supported in part by the Key Program of the National Natural Science Foundation of China (NSFC) under Grant No. 72495131; the Shenzhen Stability Science Program 2023; the Shenzhen Science and Technology Program No. JCYJ20250604141038013; and the Longgang District Key Laboratory of Intelligent Digital Economy Security.

References

Marcel Binz, Elif Akata, Matthias Bethge, Franziska Brändle, Fred Callaway, Julian Coda-Forno, Peter Dayan, Can Demircan, Maria K Eckstein, Noémi Éltető, et al. A foundation model to predict and capture human cognition. *Nature*, 644(8078):1002–1009, 2025.

Chengzhi Cao, Chao Yang, Ruimao Zhang, and Shuang Li. Discovering intrinsic spatial-temporal logic rules to explain human actions. *Advances in Neural Information Processing Systems*, 36:67948–67959, 2023.

Pablo Samuel Castro, Nenad Tomasev, Ankit Anand, Navodita Sharma, Rishika Mohanta, Aparna Dev, Kuba Perlin, Siddhant Jain, Kyle Levin, Noémi Éltető, et al. Discovering symbolic cognitive models from human and animal behavior. *bioRxiv*, pages 2025–02, 2025.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Ricky TQ Chen, Brandon Amos, and Maximilian Nickel. Learning neural event functions for ordinary differential equations. *arXiv preprint arXiv:2011.03902*, 2020.

Daryl J Daley and David Vere-Jones. *An introduction to the theory of point processes: volume II: general theory and structure*. Springer, 2008.

Edward De Brouwer, Jaak Simm, Adam Arany, and Yves Moreau. Gru-ode-bayes: Continuous modeling of sporadically-observed time series. *Advances in neural information processing systems*, 32, 2019.

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*, pages 1555–1564, 2016.

Shaun Gallagher. *Action and interaction*. Oxford University Press, 2020.

Kaiqiao Han, Yi Yang, Zijie Huang, Xuan Kan, Ying Guo, Yang Yang, Lifang He, Liang Zhan, Yizhou Sun, Wei Wang, et al. Brainode: Dynamic brain signal analysis via graph-aided neural ordinary differential equations. In *2024 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)*, pages 1–8. IEEE, 2024.

Alan G Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1): 83–90, 1971.

Seungyun Jeong, Yeseul Kang, Jincheol Lee, and Keemin Sohn. Variational embedding of a hidden markov model to generate human activity sequences. *Transportation research part C: emerging technologies*, 131:103347, 2021.

Junteng Jia and Austin R Benson. Neural jump stochastic differential equations. *Advances in neural information processing systems*, 32, 2019.

Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. *Advances in neural information processing systems*, 33:6696–6707, 2020.

Yiling Kuang, Chao Yang, Yang Yang, and Shuang Li. Unveiling latent causal rules: A temporal point process approach for abnormal event explanation. In *International Conference on Artificial Intelligence and Statistics*, pages 2935–2943. PMLR, 2024.

Guopeng Li, Yiru Jiao, Victor L Knoop, Simeon C Calvert, and JWC Van Lint. Large car-following data based on lyft level-5 open dataset: Following autonomous vehicles vs. human-driven vehicles. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pages 5818–5823. IEEE, 2023.

Shuang Li, Lu Wang, Ruizhi Zhang, Xiaofu Chang, Xuqin Liu, Yao Xie, Yuan Qi, and Le Song. Temporal logic point processes. In *International Conference on Machine Learning*, pages 5990–6000. PMLR, 2020.

Shuang Li, Mingquan Feng, Lu Wang, Abdelmajid Essofi, Yufeng Cao, Junchi Yan, and Le Song. Explaining point processes by learning interpretable temporal logic rules. In *International Conference on Learning Representations*, 2021.

Haitao Lin, Lirong Wu, Guojiang Zhao, Pai Liu, and Stan Z Li. Exploring generative neural temporal point process. *arXiv preprint arXiv:2208.01874*, 2022.

Nazanin Mehrasa, Akash Abdu Jyothi, Thibaut Durand, Jiawei He, Leonid Sigal, and Greg Mori. A variational auto-encoder model for stochastic point processes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3165–3174, 2019.

Hongyuan Mei and Jason M Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. *Advances in neural information processing systems*, 30, 2017.

Hongyuan Mei, Guanghui Qin, Minjie Xu, and Jason Eisner. Neural datalog through time: Informed temporal modeling via logical specification. In *International Conference on Machine Learning*, pages 6808–6819. PMLR, 2020.

Zhen Pan, Zhenya Huang, Defu Lian, and Enhong Chen. A variational point process model for social event sequences. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 173–180, 2020.

Jakob Gulddahl Rasmussen. Lecture notes: Temporal point processes and the conditional intensity function. *arXiv preprint arXiv:1806.00221*, 2018.

Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. *Advances in neural information processing systems*, 32, 2019.

Luciano Serafini and Artur d'Avila Garcez. Logic tensor networks: Deep learning and logical reasoning from data and knowledge. *arXiv preprint arXiv:1606.04422*, 2016.

Yanming Wan, Jiayuan Mao, and Josh Tenenbaum. Handmethat: Human-robot communication in physical and social environments. *Advances in Neural Information Processing Systems*, 35:12014–12026, 2022.

Huandong Wang, Qizhong Zhang, Yuchen Wu, Depeng Jin, Xing Wang, Lin Zhu, and Li Yu. Synthesizing human trajectories based on variational point processes. *IEEE Transactions on Knowledge and Data Engineering*, 36(4):1785–1799, 2023.

Siqiao Xue, Xiaoming Shi, James Zhang, and Hongyuan Mei. Hypro: A hybridly normalized probabilistic model for long-horizon prediction of event sequences. *Advances in Neural Information Processing Systems*, 35:34641–34650, 2022.

Siqiao Xue, Yan Wang, Zhixuan Chu, Xiaoming Shi, Caigao Jiang, Hongyan Hao, Gangwei Jiang, Xiaoyun Feng, James Zhang, and Jun Zhou. Prompt-augmented temporal point process for streaming event sequence. *Advances in Neural Information Processing Systems*, 36:18885–18905, 2023.

Ruixuan Yan, Yunshi Wen, Debarun Bhattacharjya, Ronny Luss, Tengfei Ma, Achille Fokoue, and Anak Agung Julius. Weighted clock logic point process. In *The Eleventh International Conference on Learning Representations*, 2023.

Chao Yang, Lu Wang, Kun Gao, and Shuang Li. Reinforcement logic rule learning for temporal point processes. *arXiv preprint arXiv:2308.06094*, 2023.

Chao Yang, Shuting Cui, Yang Yang, and Shuang Li. Evolving minds: Logic-informed inference from temporal action patterns. In *Forty-second International Conference on Machine Learning*, 2025.

Chenghao Yang, Hongyuan Mei, and Jason Eisner. Transformer embeddings of irregularly spaced events and their participants. *arXiv preprint arXiv:2201.00044*, 2021.

Yang Yang, Chao Yang, Boyang Li, Yinghao Fu, and Shuang Li. Neuro-symbolic temporal point processes. *arXiv preprint arXiv:2406.03914*, 2024.

Qiang Zhang, Aldo Lipani, Omer Kirnap, and Emine Yilmaz. Self-attentive hawkes process. In *International conference on machine learning*, pages 11183–11193. PMLR, 2020.

Yizhou Zhang, Karishma Sharma, and Yan Liu. Vigdet: Knowledge informed neural temporal point process for coordination detection on social media. *Advances in Neural Information Processing Systems*, 34:3218–3231, 2021.

Mark Zolotas and Yiannis Demiris. Disentangled sequence clustering for human intention inference. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9814–9820. IEEE, 2022.

Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. Transformer hawkes process. In *International conference on machine learning*, pages 11692–11702. PMLR, 2020.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model.
   [Yes] We present our model, along with the mathematical setting and assumptions, in Sec. 3. The theoretical analysis and accompanying algorithm are detailed in Sec. 4.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm.
   [Yes] The analysis of the properties and complexity is provided in Sec. 4.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries.
   [Yes] The source code is publicly available at `https://github.com/shennnnwj/flow-reasoner`.
2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results.
   [Yes] We demonstrated the full set of assumptions of all theoretical results in Sec. 3.
   (b) Complete proofs of all theoretical results.
   [No]
   (c) Clear explanations of any assumptions.
   [Yes] The clear explanations of any assumptions is provided in Sec. 4.
3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL).
   [Yes] We provide the public code at `https://github.com/shennnnwj/flow-reasoner`,

and include dataset descriptions, experimental settings, and additional implementation details in Sec. 6 and the supplementary material.

(b) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times).
[Yes] The computation of the evaluation metrics are presented in Sec. 6.

(c) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider).
[Yes] The details of computing infrastructure used is provided in Sec. 6.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

(a) Citations of the creator If your work uses existing assets.
[Yes] We use the existing datasets for experiment which are clarified in Sec. 6.

(b) The license information of the assets, if applicable.
[Not Applicable]

(c) New assets either in the supplemental material or as a URL, if applicable.
[Not Applicable]

(d) Information about consent from data providers/curators.
[Yes] The information about consent from data providers/curators is provided in Sec. 6.

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [No] We use synthetic dataset and open source dataset to conduct experiment without any sensible content.

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots.
[Not Applicable]

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable.
[Not Applicable]

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation.
[Not Applicable]

Deliberate-When-Needed: Flow-Reasoner for Neuro-Symbolic Continuous Thought — Supplementary Materials

A Algorithms

In Alg. 2, we provide the pseudocodes for thinning algorithm.

Algorithm 2 Sampling the Next Event under Threshold-Triggered Reasoning

Require: Event history $\mathcal{H}_{t_0} = \{(t_i, a_i)\}_{i=1}^{n}$, current time t_0, maximum time horizon T, current latent mental state $M(t_0)$, current reasoning instant ρ, cached ODE drive $S^{\langle\rho\rangle}$, cached action boosts $B^{\langle\rho\rangle}$, rule set $\mathcal{R}$
Ensure: Next event (t^*, a^*) where $t^* \in (t_0, T]$, $a^* \in \mathcal{A}$, or $\varnothing$ if no event occurs

$t \leftarrow t_0$
while $t < T$ **do**
  for $k \in \mathcal{A}$ **do**
    $\lambda_k \leftarrow b_k + B_k^{\langle\rho\rangle}$
  end for
  $\lambda \leftarrow \sum_{k\in\mathcal{A}} \lambda_k$
  $s \sim \mathrm{Exp}(\lambda)$
  $t' \leftarrow t + s$
  if $t' > T$ **then**
    return $\varnothing$
  end if
  if any latent mental state crosses the threshold in (t, t') **then**
    $\rho_c \leftarrow$ first threshold-crossing time in (t, t')
    evolve $M(\rho_c^-)$ from $M(t)$ using Eq. (6) with $S^{\langle\rho\rangle}$
    run multi-hop forward chaining at ρ_c to obtain rule scores $\{g_f(\rho_c)\}_{f\in\mathcal{R}}$
    update $S^{\langle\rho_c\rangle}$ and $B^{\langle\rho_c\rangle}$ from $\{g_f(\rho_c)\}$
    compute post-reasoning state $M(\rho_c^+)$
    $t \leftarrow \rho_c$, $\rho \leftarrow \rho_c$
    continue
  else
    $p_k \leftarrow \lambda_k/\lambda$ for all $k \in \mathcal{A}$
    $a^* \sim \mathrm{Categorical}(\{p_k\}_{k\in\mathcal{A}})$
    return (t', a^*) ▷ A new reasoning instant is triggered at t'
  end if
end while
return $\varnothing$

B Experimental Details

B.1 Dataset Descriptions

We evaluate our approach on both synthetic and real-world datasets:

- *Synthetic Dataset*: Generated event sequences with known ground-truth logic rules to evaluate rule recovery capability.
- *Car-Following*: Real-world driving behavior sequences capturing cognitive decision-making processes.
- *Hand-Me-That*: Human-robot interaction sequences for household manipulation tasks.

The defined predicates and logic rules for each dataset are shown in Tables 6–12. Table 5 summarizes the key statistics. For all datasets, we use a fixed random seed to shuffle all sequences and then select the specified number of training, validation, and test sequences in order from the shuffled set. In rule design, when a rule's

body contains multiple predicates, the rule type (A→M, M→M, or M→A) is determined by the type of the first predicate in the body.

Table 5: Comprehensive statistics of all datasets used in experiments.

Category	Statistics	Dataset		
		Syn Data-1	**Car-Following**	**Hand-Me-That**
Data Scale	Sequences	2,000	5000	500
	Avg. Sequence Length	7.0	3.6	30.5
	Seq. Length Std.	3.15	0.92	15.68
	Seq. Length Range	[1, 54]	[3, 5]	[4, 83]
	Avg. Time Horizon (s)	15.00	7.20	29.53
	Time Horizon Std. (s)	2.78	4.90	15.68
	Time Horizon Range (s)	[0.04, 14.52]	[0.10, 37.70]	[3.00, 82.00]
Predicates & Rules	Action Types	4	5	11
	Mental States	5	4	5
	Total Rules	11	19	22
	A→M Rules	5	7	5
	M→M Rules	2	2	6
	M→A Rules	4	10	11
	Train/Val/Test Split	1600/200/200	4000/500/500	400/50/50

B.1.1 Synthetic Dataset (Syn Data-1)

Dataset. We construct a synthetic benchmark to assess learning in a controlled setting where the underlying symbolic structure is known. The dataset contains 2,000 event sequences within a fixed time horizon of 15 seconds. Each sequence consists of action events only, with an average length of 7.0 actions (range: [1, 54]).

Predicates. We use a small set of mental predicates $\{m_j\}$ and action predicates $\{a_k\}$ (Table 6). Mental predicates represent latent continuous states, while action predicates correspond to observable event types.

Table 6: Defined predicates and explanations for the Synthetic dataset.

Category	Predicates	Explanation
Mental State	m_1	Mental State 1
	m_2	Mental State 2
	m_3	Mental State 3
	m_4	Mental State 4
	m_9	Mental State 9
Action	a_5	Action 5
	a_6	Action 6
	a_7	Action 7
	a_8	Action 8

Symbolic rules. Sequence generation uses a *ground-truth* rule set with explicit temporal relations (Table 7). During training, the model is provided with a separate *candidate* rule pool (Table 8) that removes temporal operators and includes additional rules, reflecting imperfect symbolic knowledge. The model learns rule weights from data and can down-weight or prune ineffective candidates.

Data generation. We generate sequences by simulating a temporal point process whose action intensities are modulated by latent mental states and rule-induced boosts. Mental states evolve according to the ODE dynamics in Eq. 5 with positive decay rates, and are updated at reasoning instants. At each reasoning instant (action arrival or when a mental state crosses the trigger threshold), forward chaining is applied to update rule

Table 7: Ground-truth rules used to generate the Synthetic dataset. Temporal relations are expressed over predicate timestamps $t(\cdot)$. These oracle rules are used *only* for data generation.

Rule ID	Rule Content	Temporal constraint
Action-to-Mental (A→M) rules		
GT-1	$m_4 \leftarrow a_5 \wedge a_7$	$t(a_5) = t(a_7)$
GT-2	$m_2 \leftarrow a_7 \wedge m_4$	$t(a_7) < t(m_4)$
GT-3	$m_1 \leftarrow a_8 \wedge a_6$	–
GT-4	$m_3 \leftarrow a_5 \wedge a_6$	$t(a_5) > t(a_6)$
Mental-to-Mental (M→M) rules		
GT-5	$m_2 \leftarrow m_1 \wedge a_6$	–
Mental-to-Action (M→A) rules		
GT-6	$a_5 \leftarrow m_1 \wedge m_2$	$t(m_1) < t(m_2)$
GT-7	$a_6 \leftarrow m_2 \wedge m_3$	$t(m_2) > t(m_3)$
GT-8	$a_7 \leftarrow m_1 \wedge m_3$	$t(m_1) > t(m_3)$
GT-9	$a_8 \leftarrow m_3 \wedge m_4$	–

Table 8: Training candidate rule pool for the Synthetic dataset. These rules are provided to the model during training (without temporal operators) and may differ from the oracle data-generation rules.

Rule ID	Rule Content
Action-to-Mental (A→M) candidates	
C-1	$m_4 \leftarrow a_5 \wedge a_7$
C-2	$m_2 \leftarrow a_7 \wedge m_4$
C-3	$m_1 \leftarrow a_8 \wedge a_6$
C-4	$m_3 \leftarrow a_5 \wedge a_6$
C-5	$m_9 \leftarrow a_6$
Mental-to-Mental (M→M) candidates	
C-6	$m_2 \leftarrow m_1 \wedge a_6$
C-7	$m_9 \leftarrow m_2$
Mental-to-Action (M→A) candidates	
C-8	$a_5 \leftarrow m_1 \wedge m_2$
C-9	$a_6 \leftarrow m_2 \wedge m_3$
C-10	$a_7 \leftarrow m_1 \wedge m_3$
C-11	$a_8 \leftarrow m_1$

activations, refresh the cached drives for the ODE, and compute piecewise-constant action intensities for the next interval.

Training and evaluation protocol. During training and evaluation, only the action event stream (types and timestamps) is observed; mental states are treated as latent and are not provided as supervision. The model is trained by maximizing the marked-TPP likelihood using the analytic survival decomposition under piecewise-constant intensities. This setting evaluates whether the model can infer latent mental dynamics from actions and leverage candidate symbolic rules to improve next-event type and time prediction.

B.1.2 Car-Following Dataset

Dataset Overview. The Car-Following dataset (Li et al., 2023) captures real-world driving behavior sequences. After preprocessing, we obtain 10,042 car-following episodes in total. For all experiments in this paper, we use a fixed subset of 5,000 sequences, which is split into 4,000/500/500 for training, validation, and testing, respectively. Unless otherwise noted, the statistics and experimental results reported in the main text correspond to this 5,000-sequence subset.

Sources of symbolic rules. Rules are not hand-crafted; they are proposed by an LLM from the action sequences and brief domain descriptions, yielding interpretable A→M, M→M, and M→A clauses. These are treated as candidates whose influence is fully learned; unhelpful rules are automatically down-weighted. For example, in the car-following case, given the observed actions and the domain description "drivers may become aggressive after acceleration", the LLM may propose a candidate rule such as Accelerate→DriverMood; this rule only becomes influential if its learned weight remains large after training.

Table 9: Defined predicates and explanations for Car-Following Dataset.

Predicates	**Explanation**
SafetyDistance	Driver's concern for maintaining safe following distance
SpeedUrgency	Driver's desire to increase or maintain high speed
DriverMood	Driver's affective state (aggressive vs. calm)
ConservativeTendency	Driver's inclination toward cautious behavior
Following	Maintaining consistent distance behind leading vehicle
Cruise	Maintaining constant speed without significant changes
Accelerate	Increasing vehicle speed
Decelerate	Reducing vehicle speed
Fast	Driving at high speed relative to traffic flow

Data Preprocessing. Raw trajectory data is segmented into car-following episodes using threshold-based detection. Each episode is defined as a continuous period where the ego vehicle maintains a following relationship with a leading vehicle. Timestamps are normalized relative to episode start time. After preprocessing, 10,042 episodes are obtained in total; from these, we select a fixed subset of 5,000 sequences for all experiments and split them into 4,000/500/500 for training, validation, and testing.

Table 10: Logic rules for Car-Following Dataset.

Rule ID	**Rule Content**
Action-to-Mental (A→M) Rules	
Rule-1	SafetyDistance ← Decelerate
Rule-2	SpeedUrgency ← Accelerate
Rule-3	SpeedUrgency ← Fast
Rule-4	DriverMood ← Accelerate
Rule-5	DriverMood ← Fast
Rule-6	ConservativeTendency ← Decelerate
Rule-7	ConservativeTendency ← Cruise
Mental-to-Mental (M→M) Rules	
Rule-8	ConservativeTendency ← SafetyDistance
Rule-9	DriverMood ← SpeedUrgency
Mental-to-Action (M→A) Rules	
Rule-10	Following ← SafetyDistance
Rule-11	Decelerate ← SafetyDistance
Rule-12	Accelerate ← SpeedUrgency
Rule-13	Fast ← SpeedUrgency
Rule-14	Accelerate ← DriverMood
Rule-15	Fast ← DriverMood
Rule-16	Decelerate ← ConservativeTendency
Rule-17	Cruise ← ConservativeTendency
Rule-18	Following ← ConservativeTendency

B.1.3 Hand-Me-That Dataset

Dataset Overview. The Hand-Me-That dataset (Wan et al., 2022) contains human-robot interaction sequences for household manipulation tasks. We focus on change-state episodes with dynamic transitions between task states. After preprocessing, we obtain 503 sequences in total. For all experiments in this paper, we use a fixed subset of 500 sequences, which is split into 400/50/50 for training, validation, and testing, respectively. Unless otherwise noted, the statistics and experimental results reported in the main text correspond to this 500-sequence subset.

Table 11: Defined predicates and explanations for Hand-Me-That Dataset.

Predicates	**Explanation**
TaskAwareness	Agent's awareness of needing to complete a task
ObjectInterest	Agent's interest in a particular object
SpatialPlanning	Agent's planning of movement paths in space
ToolSelection	Agent's selection of appropriate tools
CompletionUrgency	Agent's sense of urgency to complete the task
MoveTo	Moving to a location or near an object
PickUp	Picking up an object from a location
Put	Placing an object at a location or into a container
ToggleOn	Turning on an electronic device
Soak	Soaking an object in water
Open	Opening a container
Clean	Cleaning an object or location
Cool	Cooling food in a refrigerator
Slice	Slicing an object using a knife
Heat	Heating food using a heating device
Close	Closing a container

Data Preprocessing. From the original dataset, we extract change-state episodes by identifying segments where the task state undergoes transitions. Sequences shorter than 5 actions or longer than 100 actions are filtered out to ensure quality. Timestamps are normalized relative to episode start time. After preprocessing, 503 sequences are obtained in total; from these, we select a fixed subset of 500 sequences for all experiments and split them into 400/50/50 for training, validation, and testing.

B.2 Baseline Descriptions

In this paper, we primarily focus on baselines from three different fields: neural temporal point process model, logic-based model, and generative model. Below, we provide a detailed introduction to these baselines.

Neural Temporal Point Process Model

- RMTPP (Du et al., 2016): This model represents the intensity function as a nonlinear function of the event history. It uses a recurrent neural network to learn a representation of historical influences—including past events and intervals—to model the intensity.
- THP (Zuo et al., 2020): The model employs self-attention to encode event history into hidden representations for discrete timestamps, which parameterize a continuous-time intensity function. This approach efficiently captures long-term dependencies and admits structural knowledge, outperforming RNN-based methods in both computational efficiency and predictive accuracy.
- PromptTPP (Xue et al., 2023): This approach integrates a continuous-time retrieval prompt pool with a base TPP to enable sequential learning from event streams without buffering past examples. The architecture comprises the base model, the prompt pool, and a prompt-event interaction layer, collectively improving performance on streaming event sequence modeling.
- HYPRO (Xue et al., 2022): This model decouples long-horizon forecasting into two steps: proposal generation by an autoregressive TPP and likelihood-based refinement via an energy function. By selecting the most realistic predictions, it reduces cascading errors, thereby achieving superior long-term accuracy.

Table 12: Logic rules for Hand-Me-That Dataset.

Rule ID	Rule Content
Mental-to-Mental (M→M) Rules	
Rule-1	ObjectInterest ← TaskAwareness
Rule-2	SpatialPlanning ← ObjectInterest
Rule-3	ToolSelection ← SpatialPlanning
Rule-4	CompletionUrgency ← ToolSelection
Rule-5	SpatialPlanning ← TaskAwareness ∧ ObjectInterest
Rule-6	CompletionUrgency ← SpatialPlanning ∧ ToolSelection
Action-to-Mental (A→M) Rules	
Rule-7	TaskAwareness ← MoveTo
Rule-8	ObjectInterest ← PickUp
Rule-9	SpatialPlanning ← Put
Rule-10	ToolSelection ← ToggleOn
Rule-11	CompletionUrgency ← Soak
Mental-to-Action (M→A) Rules	
Rule-12	MoveTo ← TaskAwareness
Rule-13	PickUp ← ObjectInterest
Rule-14	Put ← SpatialPlanning
Rule-15	ToggleOn ← ToolSelection
Rule-16	Soak ← CompletionUrgency
Rule-17	Open ← TaskAwareness
Rule-18	Clean ← SpatialPlanning
Rule-19	Cool ← CompletionUrgency
Rule-20	Slice ← ToolSelection
Rule-21	Heat ← CompletionUrgency
Rule-22	Close ← SpatialPlanning

Logic-Based Model

- TELLER (Li et al., 2021): The model discovers temporal logic rules from noisy event sequences by formulating rule learning as a maximum likelihood problem. To address the combinatorial search space, it employs a tractable branch-and-price algorithm that alternates between generating candidate rules and evaluating their significance, efficiently identifying a concise set of predictive rules within a time budget.
- CLNN (Yan et al., 2023): This model learns interpretable Weighted Clock Logic (wCL) formulas to model how events promote or inhibit each other over time. Unlike conventional methods that rely on combinatorial optimization, CLNN introduces a continuous relaxation of the discrete rule space using smooth activation functions. This enables efficient gradient-based learning of highly expressive temporal relations.
- STLR (Cao et al., 2023): This approach explains human actions by learning spatiotemporal logic rules via a transformer-based generator and a reasoning evaluator.

Generative Model

- AVAE (Mehrasa et al., 2019): The model addresses the modeling of asynchronous action sequences by integrating a recurrent VAE with a dynamic prior network. The encoder processes the history of actions and timings into a latent distribution, from which a sample is decoded to jointly predict the next action and its inter-arrival time, moving beyond the limitations of a fixed prior.
- GNTPP (Lin et al., 2022): The model presents a modular framework for neural temporal point processes. It flexibly combines encoders (RNN- or attention-based) with a broad family of generative decoders—including diffusion, VAE, GAN, and score-based models—to approximate event time distributions. This composability makes GNTPP a highly versatile and expressive generative model for continuous-time sequences
- VEPP (Pan et al., 2020): This model introduces a VAE framework for probabilistic event sequence modeling. By encoding event histories with an LSTM and learning the latent distribution, it effectively captures

complex dependencies in inter-arrival times and event types.

- STVAE (Wang et al., 2023): The model addresses the challenge of modeling human trajectories with continuous time, variable length, and multi-faceted context. It achieves this by unifying classical temporal point processes with modern neural variational inference.

B.3 More Experiments

B.3.1 Scalability and Time Efficiency

This section provides comprehensive experimental analysis to validate the effectiveness and efficiency of our proposed Flow-Reasoner model. We examine how model performance and training characteristics scale with varying training set sizes on the synthetic dataset.

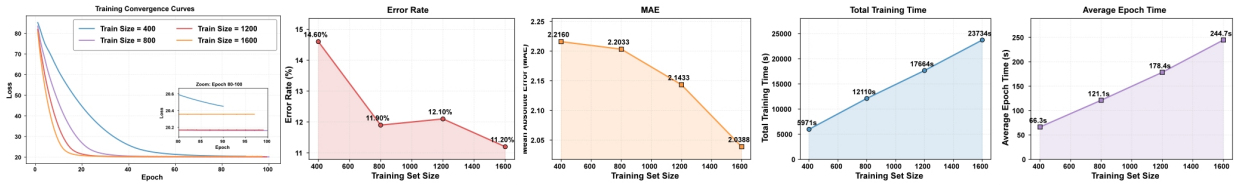

Figure 4: Comprehensive analysis of convergence behavior, model performance, and training efficiency across different training set sizes (400-1600 samples) on the synthetic dataset.

Convergence Analysis. Figure 4 shows the training convergence curves with a detailed inset focusing on epochs 80-100. The results demonstrate that larger training sets lead to both faster convergence and lower final loss values. Specifically, training with 400 samples converges to 20.45, while 1600 samples achieve the lowest loss of 20.15. Interestingly, training with 800 samples (20.35) and 1200 samples (20.2) exhibit very similar final loss values, both performing slightly worse than the 1600-sample configuration. This suggests that while larger datasets generally improve performance, there may be diminishing returns beyond a certain dataset size threshold. The convergence speed also improves significantly with larger datasets, with 1600-sample training reaching near-optimal performance by epoch 20, compared to epoch 60 for 400-sample training.

Computational Scaling. The training time analysis (top-middle and top-right plots) reveals linear scaling behavior. Total training time increases from 6,191s (400 samples) to 23,734s (1600 samples), while average epoch time scales from 61.9s to 244.7s. This linear relationship indicates efficient and predictable computational requirements, with no significant overhead penalties for larger datasets.

Performance Metrics. The accuracy analysis (bottom plots) demonstrates consistent performance improvements with larger training sets. Top-3 Error Rate decreases from 14.60% (400 samples) to 11.20% (1600 samples), with a slight fluctuation at 1,200 samples (12.10%). Mean Absolute Error shows monotonic improvement from 2.2263 to 2.0388, indicating that the model effectively leverages additional training data for enhanced predictive accuracy.

Key Insights. The comprehensive analysis reveals four critical findings: (1) *Faster Convergence:* Larger datasets accelerate both convergence speed and final performance. (2) *Linear Computational Scaling:* Training time scales predictably with data size, indicating efficient implementation. (3) *Monotonic Performance Gains:* Model accuracy consistently improves with more training data. (4) *Cost-Benefit Trade-off:* While computational cost increases linearly, the performance gains justify the additional resources, particularly for applications requiring high accuracy.

B.3.2 Ablation Study

Experiment Setting We report an ablation study on the Synthetic dataset to quantify the contribution of four design components: ODE dynamics (Non-ODE vs. ODE), symbolic reasoning depth (0-step / 1-step / 3-step forward chaining), logic boost (No Boost vs. Boost), and the event thresholding level (Low / Mid / High). Unless explicitly varied, each experiment uses the same training protocol and hyperparameters as the main baseline comparisons (Table 16). To reduce computational cost, the ablation study is conducted on a smaller training split than the main experiments; therefore, absolute metrics may differ from those in the baseline comparison table, while the relative trends across configurations remain informative for assessing module effects.

Table 13: Ablation study on the Synthetic dataset (Appendix: full configuration). We vary four modules: (i) **ODE Dynamic**: *Non-ODE* (simple time variation with learnable parameters) vs. *ODE* (closed-form ODE dynamics, our default). (ii) **Neuro-Symbolic Reasoning**: *0-step* (no forward chaining), *1-step*, and *3-step* forward chaining (our default). (iii) **Logic Boost**: *No Boost* (disable logic-strength enhancement) vs. *Boost* (enable logic-strength enhancement, our default). (iv) **Event Thresholding** (upper trigger threshold): *Low*, *Mid* (our default), and *High*. We report ER% (lower is better), MAE (lower is better), training loss, and average training time per epoch.

ID	Ablation Settings				Metrics			
	ODE Dynamic	Reasoning	Logic Boost	Thresholding	ER% ↓	MAE ↓	Loss ↓	Time ↓
Exp-1	Non-ODE	0-step	No Boost	Mid	13.24	2.24	11.80	97.05
Exp-2	ODE	0-step	No Boost	Mid	12.96	2.19	10.13	90.97
Exp-3	ODE	0-step	Boost	Mid	12.68	2.14	10.75	108.54
Exp-4	ODE	1-step	Boost	Mid	12.56	2.11	10.30	120.40
Exp-5	ODE	3-step	Boost	Mid	10.32	2.03	9.29	122.03
Exp-6	Non-ODE	0-step	No Boost	Low	11.83	2.13	11.46	105.43
Exp-7	Non-ODE	0-step	No Boost	High	12.69	2.11	11.40	105.79
Exp-8	Non-ODE	1-step	No Boost	Mid	11.94	2.24	11.68	107.35
Exp-9	Non-ODE	0-step	Boost	Mid	13.06	2.21	11.25	94.07
Exp-10	Non-ODE	3-step	Boost	Mid	12.12	2.18	11.47	113.06
Exp-11	ODE	3-step	Boost	Low	10.32	2.08	10.24	125.78
Exp-12	ODE	3-step	No Boost	Mid	11.24	2.08	11.46	105.43

Ablation Analysis Table 13 summarizes 12 configurations (Exp-1 to Exp-12). We highlight the main effects through controlled comparisons.

ODE dynamics. Comparing Exp-1 (Non-ODE, 0-step, No Boost, Mid) with Exp-2 (ODE, 0-step, No Boost, Mid) shows that replacing simple time variation with closed-form ODE dynamics improves both accuracy and efficiency: ER% decreases from 13.24 to 12.96, MAE decreases from 2.24 to 2.19, loss decreases from 11.80 to 10.13, and average epoch time reduces from 97.05s to 90.97s.

Reasoning steps. With ODE dynamics and logic boost enabled at a fixed threshold (Mid), increasing reasoning depth yields consistent gains. Exp-3 (ODE, 0-step, Boost, Mid) → Exp-4 (ODE, 1-step, Boost, Mid) yields a modest improvement (ER% 12.68 → 12.56, MAE 2.14 → 2.11). Further increasing to 3-step reasoning in Exp-5 (ODE, 3-step, Boost, Mid) brings a substantial gain (ER% 10.32, MAE 2.03), indicating that deeper forward chaining is most beneficial when coupled with the full model components.

Logic boost. Comparing Exp-12 (ODE, 3-step, No Boost, Mid) with Exp-5 (ODE, 3-step, Boost, Mid) shows that enabling logic boost is crucial under high-capacity reasoning: ER% improves from 11.24 to 10.32 and loss drops from 11.46 to 9.29. This suggests logic boost stabilizes multi-step symbolic reasoning by strengthening informative rule activations and mitigating overfitting to spurious chains.

Event thresholding. Threshold level affects both prediction quality and computation through the frequency of triggered reasoning. Under a lighter configuration (Non-ODE, 0-step, No Boost), Exp-6 (Low) achieves lower ER% than Exp-1 (Mid) and Exp-7 (High), implying that a lower trigger threshold can be advantageous when reasoning depth is minimal. Under the full configuration, Exp-5 (Mid) and Exp-11 (Low) achieve comparable ER% (both 10.32), while Exp-5 attains lower MAE (2.03 vs. 2.08) and lower loss (9.29 vs. 10.24), suggesting that a medium threshold provides a favorable trade-off between timely reasoning and avoiding excessive triggering.

Overall. Exp-5 (ODE, 3-step, Boost, Mid) is the best-performing configuration in this study (ER%=10.32, MAE=2.03, loss=9.29). Relative to Exp-1 (Non-ODE, 0-step, No Boost, Mid), it reduces ER% by 22.1% and improves MAE by 9.4%, supporting the integrated design principle of combining continuous ODE dynamics with multi-step symbolic reasoning and adaptive logic enhancement.

B.3.3 Efficiency of Deliberate-When-Needed Reasoning

We employ a deliberate-when-needed reasoning strategy, where forward chaining (FC) is executed only at reasoning instants triggered by action arrivals or when latent mental states cross the trigger thresholds, instead of running symbolic inference continuously. To quantify the computational benefit of this design, we compare two

Table 14: Time-cost and performance comparison between conditional (deliberate-when-needed) and always-on reasoning on the synthetic dataset (train/val/test = 160/20/20).

Metric	Conditional (ours)	Always reasoning
Total FC Time (s)	6.55 ± 0.52	1229.27 ± 16.65
FC Calls	351 ± 24	65,185 ± 677
FC Steps	6,275 ± 406	1,171,819 ± 11,103
Top-1 ER% ↓	74.30 ± 0.90	74.19 ± 0.91
Top-3 ER% ↓	19.46 ± 0.50	19.46 ± 0.50
MAE ↓	2.2139 ± 0.0788	2.2133 ± 0.0821

Table 15: Robustness to partial and noisy rule sets on the synthetic dataset. "Mask" randomly removes a fraction of rules, while "Corrupt" perturbs rule bodies (e.g., by replacing body predicates).

Condition	Level	Top-3 ER% ↓	MAE ↓
Full rules	0%	10.31	2.20
Mask (partial rules)	30%	14.60	2.04
Mask (partial rules)	60%	19.34	2.28
Corrupt (noisy rules)	10%	12.26	2.12
Corrupt (noisy rules)	20%	18.12	2.26

reasoning strategies on the synthetic dataset: (i) **Conditional (ours)**, where FC is executed only at reasoning instants; and (ii) **Always reasoning**, where FC is executed at every time point on a fixed discretized time grid regardless of triggers, while keeping the rest of the training and evaluation pipeline unchanged. We use a smaller synthetic split (train/val/test = 160/20/20) and report both efficiency metrics (total FC time, number of FC calls, and FC steps) and prediction metrics (Top-1/Top-3 ER% and MAE). ER% is computed from Top-K accuracy (ER% = 100×(1−Top-K accuracy)), and MAE is the mean absolute error of predicted event times. As shown in Table 14, conditional triggering reduces FC calls from ∼65k to ∼351 and FC steps from ∼1.17M to ∼6.3k, yielding ∼99.5% FC time savings (≈190× speed-up), while keeping Top-K ER% and MAE essentially unchanged. This indicates that the dominant cost of symbolic inference stems from unnecessary FC executions in non-triggered regions, and that deliberate triggering removes this overhead with negligible impact on accuracy.

B.3.4 Robustness to Imperfect Rule Sets

In practical settings, symbolic rules are often imperfect: the available rule set may be incomplete (missing relevant clauses) and/or noisy (containing inaccurate or irrelevant conditions). To evaluate robustness under weak or unreliable symbolic knowledge, we conduct controlled perturbations of the rule set on the synthetic dataset and measure how predictive performance changes. Starting from the (pruned) training rule set used in the main synthetic experiment, we apply two perturbation types and re-train/evaluate under the same pipeline: (i) **Mask (incomplete rules)**, which randomly removes a fraction of rules to simulate missing knowledge; and (ii) **Corrupt (noisy rules)**, which perturbs the rule bodies (e.g., by replacing body predicates) for a fraction of rules to simulate inaccurate knowledge. All other settings remain unchanged, and we report Top-3 ER% and MAE. As summarized in Table 15, performance degrades as symbolic structure is removed or corrupted, but the degradation is gradual and the model remains functional. With incomplete rules, the full rule set yields Top-3 ER% / MAE of 10.31 / 2.20, which becomes 14.60 / 2.04 when masking 30% of rules and 19.34 / 2.28 when masking 60%. With noisy rules, corrupting 10% and 20% of rules yields 12.26 / 2.12 and 18.12 / 2.26, respectively. These results indicate that the model benefits from good symbolic structure, yet does not collapse when rules are partial or moderately noisy; instead, it degrades gracefully by relying more on its continuous ODE–TPP backbone when symbolic guidance weakens.

C Reproducibility Analysis

C.1 Computing Infrastructure

All the synthetic data and real-world data experiments, including the comparison experiments with baselines, are performed on Ubuntu 20.04.3 LTS system with Intel(R) Xeon(R) Gold 6248R CPU @ 3.00GHz, 227 Gigabyte

Table 16: Descriptions and values of hyper-parameters used for models trained on the synthetic and real-world datasets.

Module	Hyper Parameters	Value Used		
		Syn Data-1	**Car-Following**	**Hand-Me-That**
Architecture	Predicate matching temp (τ)	0.5	0.5	0.5
	Soft-AND temp (α)	0.01	0.5	0.5
	Soft-OR temp (β)	7.0	8.0	8.0
	LogSumExp temp (δ)	0.1	0.1	0.1
	Action predicate decay rate (λ)	0.5	0.5	0.5
	Forward-chaining passes (L)	3	3	3
	Forward-chaining tolerance (ϵ)	2.5×10^{-4}	5×10^{-4}	5×10^{-4}
	Threshold upper bound	0.65	0.60	0.60
	Threshold lower bound	0.05	0.05	0.05
Training	Learning rate	5×10^{-3}	3×10^{-3}	3×10^{-3}
	Batch size	32	32	32
	Number of epochs	100	100	50
	Optimizer	Adam	Adam	Adam
Regularization	Gradient clipping (max norm)	5.0	5.0	5.0
	Early Stopping Patience	15	15	15
	Early Stopping Threshold	1×10^{-4}	1×10^{-4}	1×10^{-4}

memory.

C.2 Hyper-Parameter Selection

We present the selected hyper-parameters for the synthetic dataset and the two real-world datasets in Tab. 16. The hyper-parameter selection metric is a trade-off between model performance and time efficiency.

D Rule Sources and Fairness of Comparison

A natural concern in neuro-symbolic modeling is whether performance gains come mainly from strong prior rules rather than from the learned model itself. In this section, we clarify how candidate rules are constructed in Flow-Reasoner and why the experimental comparisons remain fair.

D.1 Candidate Rules Rather than Oracle Logic

Flow-Reasoner does not assume access to oracle rules during training. In the synthetic setting, the ground-truth rules are used only to generate event sequences in the simulator. The model is instead trained with a perturbed candidate rule pool obtained by modifying the simulator rules, for example by dropping predicates, adding distractor clauses, or simplifying temporal structures. As a result, the candidate rules available during training differ from the true data-generation rules, and the model must learn which rules are useful and how strongly they should influence prediction.

In the real-world setting, candidate rules are not manually specified as fixed logic. Instead, they are proposed by a large language model (LLM) from observed action patterns together with brief domain descriptions. These rules provide only coarse and potentially imperfect symbolic hypotheses rather than ground-truth knowledge. Their influence is fully determined by learned rule weights, and unhelpful rules can be down-weighted or effectively removed during training.

D.2 Fairness of Experimental Comparison

All compared methods are trained on the same observed event sequences. Baseline models do not include a symbolic reasoning layer and therefore cannot directly use candidate rules without substantial architectural modification. We therefore compare Flow-Reasoner against strong representatives of neural temporal point process models, rule-based models, and generative baselines under the same prediction setting.

Importantly, none of the compared models, including Flow-Reasoner, has access to the true simulator rules during training. Therefore, the observed gains cannot be attributed to oracle symbolic supervision. Instead, the

results indicate that the improvements come from the learned interaction between weak symbolic candidates and continuous latent dynamics, rather than from fixed prior knowledge.

E Limitation & Broader Impacts

Our approach has strong potential for human–AI collaboration by enabling timely and interpretable inference of latent mental states and more accurate forecasting of human actions, with possible applications ranging from assistive daily living to safer autonomous driving. However, it also has several limitations. First, the decoder relies on candidate rule templates, which improve interpretability but may introduce bias and limit cross-domain transfer when the available symbolic structure is incomplete or domain-specific. Future work should therefore explore more scalable rule-mining and structure-learning methods to broaden coverage and reduce bias. Second, timeline discretization may introduce noise when sampling latent mental events and may require extensive tuning to select an appropriate resolution. This motivates future work on automated hyperparameter search and adaptive or learned time grids to better balance fidelity and efficiency. Finally, because intention inference can affect decision-making in sensitive settings, responsible deployment should include safeguards for privacy, fairness, and transparency, such as data minimization, bias audits of learned rules, and interpretable rationales for predictions.