Discovering Intrinsic Spatial-Temporal Logic Rules to Explain Human Actions

Chengzhi Cao[1,2]*, **Chao Yang**[2], **Ruimao Zhang**[2], **Shuang Li**[2]
University of Science and Technology of China,
The Chinese University of Hong Kong (Shenzhen)
chengzhicao@mail.ustc.edu.cn, 222043011@link.cuhk.edu.cn,
zhangruimao@cuhk.edu.cn lishuang@cuhk.edu.cn

Abstract

We propose a *logic-informed* knowledge-driven modeling framework for human movements by analyzing their trajectories. Our approach is inspired by the fact that human actions are usually driven by their intentions or desires, and are influenced by environmental factors such as the spatial relationships with surrounding objects. In this paper, we introduce a set of *spatial-temporal logic rules* as knowledge to explain human actions. These rules will be automatically discovered from observational data. To learn the model parameters and the rule content, we design an expectation-maximization (EM) algorithm, which treats the rule content as latent variables. The EM algorithm alternates between the E-step and M-step: in the E-step, the posterior distribution over the latent rule content is evaluated; in the M-step, the rule generator and model parameters are jointly optimized by maximizing the current expected log-likelihood. Our model may have a wide range of applications in areas such as sports analytics, robotics, and autonomous cars, where understanding human movements are essential. We demonstrate the model's superior *interpretability* and *prediction* performance on pedestrian and NBA basketball player datasets, both achieving promising results.

1 Introduction

For a human, although the exhibited movements can be complex, the logic behind the actions is usually simple, clear, and can be generalized [14]. The *logic rules* present a compact and high-level *knowledge representation*, defining what actions tend to be executed under what conditions. There has been a great interest and business value in unveiling the human logic from the observational movements and actions [15]. We provide two motivating examples below.

In *sports analytics*, understanding each player's behavior preferences or tendencies under various scenarios will provide coaches with valuable information [1]. Usually, coaches need to watch the game or training videos for hundreds of hours before they can summarize the discoveries into compact principles. *Could we design an algorithm to synthesize these principles from the raw action data automatically*? One can imagine, such a tool will significantly reduce the workload of coaches, providing more granular insight into each player's capabilities and strategies, and aiding in personalized training and match strategy design [13].

In *self-driving car*, it's essential to enable the self-driving cars to "read human's mind" like humans. This requires the self-driving cars to automatically *understand human intentions and reasoning* when they are running on the same roads with human drivers [2]. If self-driving cars can automatically distill logic rules from their observed *low-level noisy* human actions and movement trajectories, it will increase the technical reliability and accelerate the widespread use of self-driving cars.

*Work done during an internship at the Chinese University of Hong Kong (Shenzhen)

For human actions, lots of the governing rules would be regarding the *spatial-temporal relation* with the surrounding environments and their intentions [9]. For example, when a basketball player with a ball is within the scoring range, his/her action, such as shoot, pass, or triple threat, is influenced by historical and current surrounding factors, such as the current locations of the player and the defenders, the time elapses of the game, the success shooting rate of the player in today's game, and so on. The quick decision made by the player is actually reflecting a composition of all these factors, which can be described as a collection of spatial-temporal logic rules in our model.

Formally, in our introduced spatial-temporal logic rules, the logic variable (i.e., predicate) set will include spatial-temporal relation predicates, in addition to the commonly defined object property and relation predicates. The rule content will *capture the spatial relation of the object with surrounding objects, as well as the temporal ordering constraints of the events*.

Our methods have the following distinct features:

From the modeling perspective: (*i*) Our human action model is a rule-based *probabilistic* model, which treats each hidden rule as a "soft" constraint. We assume each rule will be executed by humans with probabilities, and this *tolerates the uncertainties* in data. (*ii*) Our model directly uses low-level, fine-grained, and (may) irregularly-spaced action times and locations (i.e., original *3d coordinates*) as inputs, as opposed to other rule-based models, where one needs to first extract relational data as inputs. (*iii*) Our spatial and temporal predicates are also probabilistic. For predicates such as "left of" or "before", we model them as kernel functions with learnable variables. In this way, our introduced spatial-temporal predicates are *smooth functions* of the input locations and times, which increases model flexibility.

From the learning perspective: We propose a *tractable* and *differentiable* algorithm that can jointly learn the *rule content* and *model parameters* from observational data. The learning framework is designed to maximize the likelihood of the human action trajectories. Specifically, we propose to use a neural rule generator to generate the spatial-temporal logic rule set. Our continuous rule generator parameters will be optimized in a differentiable way. The overall learning procedure is an expectation-maximization (EM) algorithm, where we treat the rule set as latent variables. In the E-step, the posterior distribution over the latent rule set is evaluated. In the M-step, both the rule generator parameters and the model parameters are optimized by maximizing the expected log-likelihood with respect to the current posterior. We demonstrated the promising performance of our model in terms of human action *prediction* and *explanation* on two interesting real datasets.

2 Related Work

Logic Rule Learning. Learning logic rules from raw data has been widely studied for various downstream tasks, such as motion inference [4] and healthcare analysis [10]. Learning rules via an exact search requires enumerating all combinations of the logic predicates and is intractable in most problems. One has to design heuristic searching algorithms by leveraging some structural properties of the problems. For example, Dash et al. [5] formulated a convex rule learning problem and proposed a column generation algorithm to expand the rule set gradually. Wang et al. [22] designed a Bayesian framework for learning rule classifiers and derived bounds on the support of rules in a MAP solution. Recently, Yang et al. [24] proposed an interesting end-to-end differentiable approach (Neural LP) to learn the parameters and structure of logical rules. Qu et al. [16], and Sadeghian et al. [19] proposed an efficient logic rule mining algorithm based on the knowledge graph data. *However, none of these advanced rule mining methods can directly work on spatial-temporal human action data when the inputs are raw event 3d coordinates and types.*

Spatio-Temporal Dynamics for Event Data. Since the human actions are irregular spatial-temporal event data, we also briefly discuss *probabilistic* models for such *event sequences*. Modeling the spatial-temporal dynamics of discrete events is foundational in many scientific fields and applications [17]. Shen et al. [20] proposed a novel deep learning model for spatial-temporal events such as taxi data and achieved promising prediction accuracy. Zhou et al. [27] integrated deep learning methods with spatiotemporal point processes and modeled the intensity function as a latent stochastic process. Chen et al. [3] deployed two novel architectures, including jump and attentive continuous-time normalizing flows, to learn the dynamics of the spatiotemporal event data. Repe et al. [18] learned canonical spatiotemporal point cloud representation using a latent ODE and continuous normalizing flows to generate shapes continuously in spacetime. *However, these spatial-temporal event models are governed by hard-to-interpret dynamic functions and cannot be generalized to*

model human action events. Could we propose a model with logic-informed dynamic functions to explain the spatial-temporal human action events?

3 Our Model

3.1 Data: Human Actions Recorded as Spatial-Temporal Event Sequences

Consider a set of objects, denoted as $\mathcal{C}$. For the object $c \in \mathcal{C}$, its trajectories and the key actions observed up to t can be summarized as a sequence of temporally ordered events

$$\mathcal{H}^c_{t-} = \{e^c_1 = (t^c_1, s^c_1, \kappa^c_1), \ldots, e^c_n = (t^c_n, s^c_n, \kappa^c_n) \mid t^c_n < t\},$$

where $t \in R^+$ is the time, $s \in R^2$ is the location, and $\kappa \in \mathcal{K}$ is the event (i.e., action) type.

3.2 Definition of Spatial-Temporal Predicates

Static Predicate Given the object set $\mathcal{C}$, the *predicate* is defined as the *property* or *relation* of objects, which is a *logic function* as follows

$$X(\cdot) : \mathcal{C} \times \mathcal{C} \cdots \times \mathcal{C} \mapsto \{0, 1\}.$$

For example, $Smokes(c)$ is a property predicate and $Friend(c, c')$ is a relation predicate.

Spatial-Temporal Predicate In our paper, we extend the above static predicates to spatial-temporal predicates, which include spatial-temporal *property* predicates and spatial-temporal *relation* predicates.

Specifically, the spatial-temporal *property* predicates are defined as

$$X(\cdot) : \mathcal{C} \times \cdots \times \mathcal{C} \times \mathcal{T} \times \mathcal{S} \mapsto \{0, 1\}.$$

For example, $PickupKey\,(c, t, s)$ is a spatial-temporal property predicate. Suppose an entity c_1 picked up the key at time t_1 in location s_1, then the predicate will be grounded as True (1) at (c_1, t_1, s_1), i.e., $PickupKey(c_1, t_1, s_1) = 1$; otherwise it is False (0).

Given the observational human action data, the grounded predicate

$$\{PickupKey\,(c, t, s)\}_{t=1,2,\ldots}$$

can be modeled as a sequence of discrete events – when the predicate becomes True, an event happens. In general, the grounded spatial-temporal property predicate $\{X(v_t)\}_{t=1,2,\ldots}$ is a discrete event sequence, where the event occurrence times and locations are irregular.

The spatial-temporal *relation* predicates are introduced to define the spatial and temporal relations of two entities. Specifically, they are defined as

$$R(\cdot, \cdot) : (\mathcal{C} \times \mathcal{T} \times \mathcal{S}) \times (\mathcal{C} \times \mathcal{T} \times \mathcal{S}) \mapsto \{0, 1\}.$$

Spatial-temporal relation predicates are logic variables indicating the spatial-temporal relations of two objects, where we further divide them into *temporal relation* predicates, *static spatial relation* predicates, and *dynamic spatial relation* predicates. More details can be found in Appendix.

It is noteworthy that all these boolean predicates can be converted to probabilistic ones. We can soften these logic functions by kernel functions with learnable parameters to tolerate uncertainties in data.

3.3 Definition of Spatial-Temporal Logic Rules

We will consider spatial-temporal logic rules where the body part contain spatial-temporal predicates as relation constraints. For example, a sensible rule will look like

$$f : Y_{\text{TurnAround}}(c, t, s) \leftarrow X_{\text{PickUpKey}}(c, t, s) \bigwedge R_{\text{InFront}}((c', t, s'), (c, t, s)) \bigwedge R_{\text{Behind}}((c'', t, s''), (c, t, s))$$

where $c \in \mathcal{C}_{\text{person}}$, $c' \in \mathcal{C}_{\text{block}}$, and $c'' \in \mathcal{C}_{\text{key}}$. In general, the *spatial-temporal logic rule* in our paper is defined as a logical connectives of predicates, including property predicates and spatial-temporal relation predicates,

$$f : Y(v) \leftarrow \bigwedge_{X_{\text{property}} \in \mathcal{X}_f} X_{\text{property}}(v') \bigwedge_{R_{\text{spatial-temporal}} \in \mathcal{R}_f} R_{\text{spatial-temporal}}(v'', v) \tag{1}$$

where $Y(v)$ is the *head predicate* evaluated at the entity-time-location triplet v, $\mathcal{X}_f$ is the set of property predicates defined in rule f, and $\mathcal{R}_f$ denotes the set of spatial-temporal relation predicates defined in rule f.

3.4 Logic-Informed Action Event Models

We consider a setting where we can fully observe the trajectories of all the moving objects, including their real-time locations and key actions (i.e., events), denoted as $\mathcal{H}_t$. We aim to propose a logic-informed spatial-temporal model to predict and explain the action type given the entity-time-location triplet $v = (c, t, s)$ (i.e., query) and $\mathcal{H}_t$.

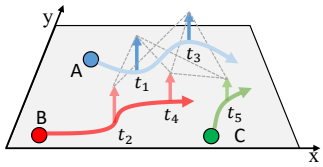


Figure 1: Illustration of feature construction using a simple logic formula with temporal relation predicate ($t_1 < t_2$), $f : Y \leftarrow A \wedge B \wedge C \wedge (A \text{ Before } B)$. The rule defines the template to gather combinations of the body predicate history events. Here predicate A has 2 events and predicate B has 1 event, the temporal relation constraint would lead to valid combinations (also called "paths"). This type of feature construction can be extended to spatial-temporal cases, where we count the valid paths as the feature.

Logic-informed feature The main idea is to construct the model features using spatial-temporal logic rules, as defined in Eq. (1). Intuitively, given the entire trajectories $\mathcal{H}_t$ and the query $v = (c, t, s)$, the body part of the rule defines the evidence to be selectively gathered from *history* to deduce the event type for query entity $v = (c, t, s)$.

Assume that for each possible event type $\kappa \in \mathcal{K}$, there exist multiple rules such as Eq. (1) to explain its occurrence, with κ being the *head predicate*.

Given an individual rule as Eq. (1), we propose to build the *feature* that is conditional on history and query as

$$\varphi_f(\kappa \,|v, \mathcal{H}_t) = \text{sign}(\kappa \in f) \cdot \sum_{\text{path} \in \{\mathcal{H}_t, v\}} g_f\,(\text{path})\,, \tag{2}$$

where we introduce a function $g_f(\cdot)$ to check the body conditions of f given a "path". We use a simple example to explain how to compute features, as shown in Figure 1. As illustrated, the feature computes the valid total number of "paths" given the data and query.

Suppose there is a rule set $\mathcal{F}_\kappa$, where the event κ is the head predicate. All the rules will play together to reason about the occurrence of κ. For each $f \in \mathcal{F}_\kappa$, one can compute the features as above. Given the rule set $\mathcal{F}_\kappa$, we model the probability of the event κ as a log-linear function of the features, i.e.,

$$p(\kappa|v, \mathcal{H}_t) \propto \exp\left(\sum\nolimits_{f \in \mathcal{F}_\kappa} w_f \cdot \phi_f(\kappa|v, \mathcal{H}_t)\right), \tag{3}$$

where $w = [w_f]_{f \in \mathcal{F}} \geq 0$ are the learnable weight parameters associated with each rule. All the model parameters can be learned by maximizing the likelihood, which can be computed using the above Eq. (3). We intend to train a rule generator p_θ and an evaluator p_w to maximize the likelihood of training data as:

$$\max_{\theta, w} \mathcal{O}(\theta, w) = \mathbb{E}_{(\kappa, v, \mathcal{H}_t)}[\log \mathbb{E}_{p_\theta}[p_w(\kappa|v, \mathcal{H}_t)]]. \tag{4}$$

More details can be found as follows.

4 Our Learning Algorithm

Our goal is to jointly learn the set of spatial-temporal logic rules $\{\mathcal{F}_\kappa\}_{\kappa \in \mathcal{K}}$ and their weights by the maximum likelihood method, where each rule has a general form as Eq. (1).

To discover each rule, the algorithm needs to navigate through the combinatorial space considering all the combinations of the property predicates and their spatial and temporal relations. To address this computational challenge, we propose a tractable (functional) EM algorithm that treats the rule set as latent variable z. The rules will be generated by a hidden neural rule generator. The overall learning framework alternates between an E-step, where the posterior distribution of the latent rule space is evaluated (rule generation), and the M-step, where the model parameters and rule generator parameters are optimized. Please refer to Fig. 2 for an illustration.

Our goal is to maximize the likelihood of the observed human action events $\{\kappa^{(i)}\}_{i=1,\ldots,n}$. Using the chain rule, we have

$$\log p_w(\{\kappa^{(i)}\}_{i=1,\ldots,n}) = \sum_{i=1}^{n} \log p_w(\kappa^{(i)} \mid v^{(i)}, \mathcal{H}_{t^{(i-1)}}). \tag{5}$$

To simplify the notation, we will use $p_w(\kappa^{(i)})$ to stand for $p_w(\kappa^{(i)} \mid v^{(i)}, \mathcal{H}_{t^{(i-1)}})$ in the following. Given a latent rule set z, we have to marginalize the posterior of z to get the above log-likelihood.

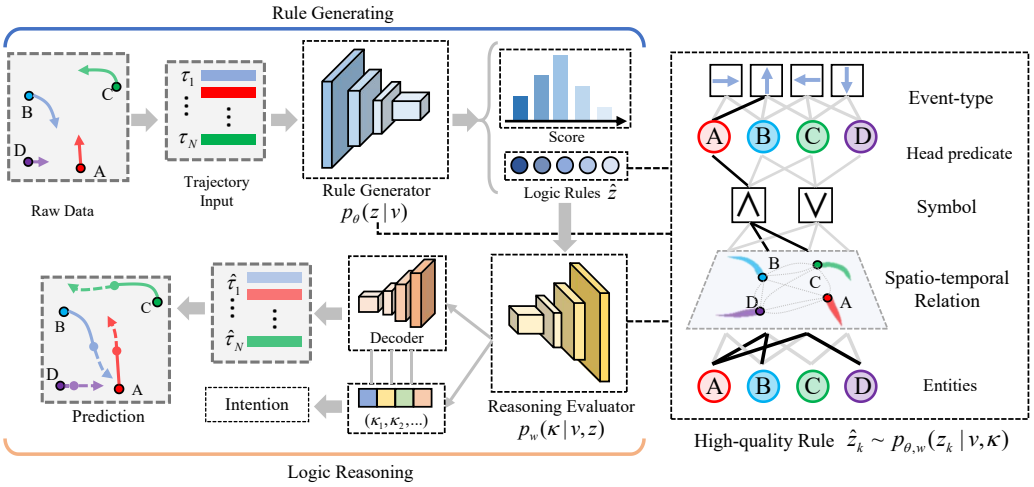


Figure 2: The overview of our proposed framework. It contains two important processes: rule generating and logic reasoning. Given the past motion of each entity on a scene over the last few seconds, the rule generator generates logic rules for the reasoning predictor. The reasoning predictor takes the generated rules as input, and predict the intention of each entity. It is optimized by EM algorithm. In the E-step, a set of top K rules are selected from all generated rules via posterior inference. Finally in the M-step, the rule generator is updated to be consistent with the high-quality rules identified in E-step.

However, the exact inference of z is intractable. We will introduce an amortized recognition network $p_\theta(z|\kappa^{(i)})$ to approximate the true posterior. We have

$$\log p_w(\kappa^{(i)}) = D_{KL}(p_\theta(z|\kappa^{(i)})||p_w(z|\kappa^{(i)})) + \mathcal{L}(\theta, w; \kappa^{(i)}), \tag{6}$$

where the first term is the KL divergence of the approximate from the true posterior, and the second term $\mathcal{L}(\theta, w; \kappa^{(i)})$ is the variational lower bound (ELBO). It can be represented as:

$$\mathcal{L}(\theta, w; \kappa^{(i)}) = -D_{KL}(p_\theta(z|\kappa^{(i)})||p_w(z)) + \mathbb{E}_{p_\theta(z|\kappa^{(i)})}[\log p_w(\kappa^{(i)}|z)]. \tag{7}$$

And $\log p_w(\kappa^{(i)}) \geq \mathcal{L}(\theta, w; \kappa^{(i)})$. The bound becomes tight when the approximate posterior matches the true one. Our goal is to optimize the variational parameters θ and model parameters w from the ELBO lower bound.

4.1 Rule Generator

We deploy Transformer-based framework to model the rule generator p_θ. We define the distribution of a set of rules as follows:

$$p_\theta(z \mid v, \mathcal{H}_t) = \Psi(z|N, \text{Trans}_\theta(v, \mathcal{H}_t)), \tag{8}$$

where $\Psi(\cdot)$ is multinomial distributions, N is the number of the top rules, and $\text{Trans}_\theta(v, \mathcal{H}_t)$ defines a distribution over compositional rules with spatial-temporal states. The generative process of the rule set is quite intuitive, where we simply generate N rules to form z. In fact, this $p_\theta(z \mid v, \mathcal{H}_t)$ is a flexible posterior approximation function, which will be optimized by the EM type algorithm.

We choose transformer over graph neural network (GNN) as our baseline because transformer architectures are based on a self-attention mechanism that is able to capture long-range relationships, as opposed to recurrent neural networks that process sequence elements recursively and can only take into account short-term context. Note that the graph operations in GNN are designed to learn node representations on the fixed and homogeneous graphs. The limitations especially become problematic when learning representations on a changeable graph that consists of various types of nodes and edges.

4.2 Rule Evaluator

Eq. (3) is our rule evaluator (suppose we know the rule content). Here we assume the rule content is latent, and the rule evaluator is given as

$$p_w(\kappa|v, z, \mathcal{H}_t) = \frac{\exp\left(\sum_{f \in z_\kappa} w_f \cdot \phi_f(\kappa|v, \mathcal{H}_t)\right)}{\sum_{\kappa'} \exp\left(\sum_{f \in z_{\kappa'}} w_f \cdot \phi_f(\kappa'|v, \mathcal{H}_t)\right)}. \tag{9}$$

4.3 Optimization

We optimize the rule generator p_θ and reasoning evaluator p_w to maximize the objective in Eq. (4). At each training iteration, we first update the reasoning predictor p_w according to some rules generated by the generator, and then update the rule generator p_θ.

In our network, the latent rule set will be automatically discovered. The best set of logic rules is approximately obtained by sampling and preserving the top-K rules according to their posterior probabilities. Specifically, as shown in Eq. (8), the posterior probabilities of the latent rule z is obtained by a Transformer type of encoder, which maps the input observed action trajectories to a latent explanatory rule space. Each candidate rule is generated in the latent rule space token-by-token (token means logic variable/predicate in our setting) in a sequential manner and meanwhile the posterior probability of each rule sequence can be evaluated. When optimizing the evaluator, we draw several rules $\hat{z}$ for each query and let the evaluator use $\hat{z}$ to predict κ. For each query, we aim to identify top K rules z_I from all generated rules $\hat{z}$, i.e., $z_I \subset \hat{z}, |z_I| = K$. It is accomplished by taking into account the posterior probabilities of each subset of logic rules z_I with prior from the rule generator p_θ and likelihood from the reasoning predictor p_w. Then, the likely set of high-quality rules can be obtained by sampling from the posterior. Specifically, when a series of rules produced from the rule generator p_θ, we calculate the weights of each rule $z^{(i)}$ as follows:

$$H(z^{(i)}) = \{p_w(\kappa|z^{(i)}) - \frac{1}{|\mathcal{A}|}\} + \log \text{Trans}_\theta(z^{(i)}|v, \mathcal{H}_t), \tag{10}$$

where $\mathcal{A}$ is the set of all candidate event type inferred by logic rules. $\text{Trans}_\theta(z^{(i)}|v, \mathcal{H}_t)$ is the probability of rule computed by the generator. For a subset of rules $z_I \subset \hat{z}$, the log-probability can be approximated as: $\log p_{\theta,w}(z_I|v, \mathcal{H}_t) \approx \sum_{z^{(i)} \in z_I} H(z^{(i)}) + \log \Psi(z_I|N, \text{Trans}_\theta(v, \mathcal{H}_t)) + \text{const}$. This equation inspired us to use the distribution $q(z_I) \propto \exp(\sum_{z^{(i)} \in z_I} H(z^{(i)}) + \log \Psi(z_I|N, \text{Trans}_\theta(v, \mathcal{H}_t)))$ as approximation of the posterior. Each rule $z^{(i)}$ sampled from $q(z_I)$ independently can be formed with N logic rules.

Clearly, $H(z^{(i)})$ can be regarded as the quality of candidate rules, with consideration of the evaluator p_w. It is calculated as the contribution of a rule to the correct event type minus the average contribution of this rule to the other candidate responses. A rule is more significant if it obtains a higher score to the correct event type and a lower score to other potential predictions.

After getting several high-quality rules from training data, we further utilize these rules to update the parameters of rule generator p_θ. Concretely, we regard the generated high-quality rules as part of training data, and update the rule generator by maximizing the log-likelihood as follows:

$$\mathcal{O}(\theta) = \log p_\theta(z_I|v, \mathcal{H}_t) = \sum_{z^{(i)} \in z_I} \log \text{Trans}_\theta(v, \mathcal{H}_t) + \text{const}. \tag{11}$$

By learning to generate high-quality rules, the rule generator will reduce the search area and produce better empirical results for the reasoning predictor.

5 Experiments

In this section, we provide some implementation details and show ablation studies as well as visualization to evaluate the performance of our framework. We compare our model with several state-of-the-art approaches, including PECNet [12], NMMP [7], STGAT [8], SOPHIE [19], STAR [25], Y-Net [11], MID [6], Social-SSL [21], and NSP-SFM [26].

5.1 Datasets

Stanford Drone Dataset. This dataset consists of more than 11,000 persons in 20 scenes captured from the campus of Stanford University in bird's eye view. We follow the [23] standard train-test split, and predict the future 4.8s (12 frames) using past 3.2s (8 frames). Note that SDD dataset does not provide explicit pedestrian's action. Instead, we record them as an abstract encoding of the pedestrian's speed and location. The action contains [left, right, straight, turn around].

NBA SportVU Dataset. It is collected by NBA using the SportVU tracking system, which reports the trajectories of the ten players and the ball in real basketball games. Each trajectory contains the 2D positions and velocities of the offensive team, consisting of 5 players. We predict the future 10 timestamps (4.0s) based on the historical 5 timestamps (2.0s). Each player's action contains [left, right, straight, turn around, pass, shoot].

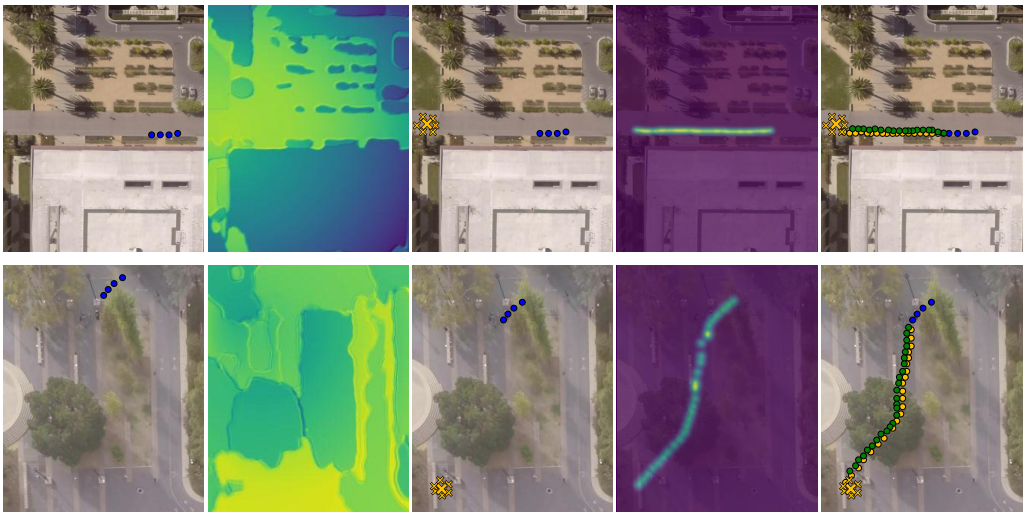


Figure 3: Visual results of our method in Trajectory Forecasting. The first column (a) shows the past observed trajectory for last 5 seconds in blue. The second column (b) shows the heatmap of predicted goal in the following 30 seconds. The third column (c) shows some sampled goals from the estimated distribution. The forth column (d) shows trajectory heatmaps. The last column (e) shows the predicted trajectories, green indicating the ground-truth trajectories and orange our predictions.

Metrics	PECNet	Y-Net	MID	NMMP	STGAT	SOPHIE	Social SSL	NSP-SFM	STAR	Ours
ADE_{20}	20.03	7.85	7.61	14.12	14.43	15.56	6.63	<u>6.52</u>	10.76	**6.41**
FDE_{20}	33.86	11.85	14.30	20.68	22.59	24.32	12.23	<u>10.61</u>	17.03	**10.23**
F1 score	0.37	0.54	0.49	0.41	0.33	0.39	0.53	<u>0.58</u>	0.56	**0.59**

Table 1: Quantitative results (ADE_{20}, FDE_{20} and F1 score) of trajectory prediction in SDD dataset. The bold/underlined font represent the best/second best result.

5.2 Metrics

Here, we adopt two metrics for evaluation: Average Displacement Error (ADE_K) and Final Displacement Error (FDE_K). Specifically, ADE_K is the minimum among K average distances of the K predicted trajectories to the ground-truths in terms of the whole trajectories. FDE_K is the minimum distance among K predicted endpoints to the ground-truth endpoints. Moreover, we also calculate the accuracy and F1 score of event-types predicted from each network.

5.3 Quantitative Analysis

We compare our method with several state-of-the-art approaches, and table 1 presents the qualitative results on the SDD dataset. The proposed model achieved the best performance in ADE, FDE and accuracy. We observe that our method significantly outperforms all baselines measured by ADE and FDE. It achieves an ADE of 6.41 and FDE of 10.23 at $K = 20$ in SDD datset, which exceeds the previous state-of-the-art performance of Y-Net [11] by 18.3% on ADE and 13.6% on FDE. In NBA dataset, our method also achieve higher performance than Y-Net. This is because Y-Net firstly assume that the waypoint lies on a straight line segment connecting the sampled goal and the past trajectory, then use a multivariate Gaussian prior centered at the assumed location. This assumption can not well suit in other complex conditions, such as the trajectory of players in the NBA dataset.

Compared with MID [6], we also obtain 15.7% ADE and 28.4% FDE improvement. Note that they carefully design a Transformer-based architecture to model the temporal dependencies in trajectories, but ignore the spatial correlation of agents. Our transformer-based network aims at generating high-quality logic rules based on spatial-temporal relation under the principle to maximize the likelihood of the observational human actions. For NSP-SFM, it obtains high performance in SDD dataset but can not achieve the same level in NBA dataset. It combines physics with deep learning for trajectory prediction and accommodates arbitrary physics models. The limitation of it lies in specific physics models, such as pedestrians, and is deterministic. So it can not deal with some strategy-based

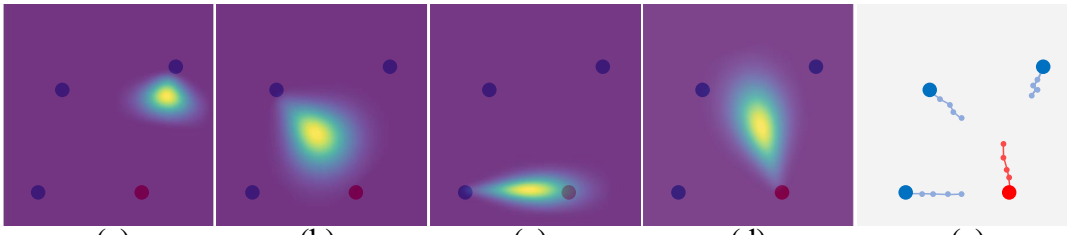


Figure 4: Estimated distributions for each player's intention in cartesian space.

Times	Y-Net	MID	NSP-SFM	Social-SSL	Social Implicit	Social VAE	ABC+	Ours
1.0s	<u>0.38/0.48</u>	0.45/0.59	0.41/0.52	0.48/0.61	0.45/0.53	0.49/0.66	0.45/0.56	**0.30/0.40**
2.0s	<u>0.63/0.93</u>	0.76/1.06	0.67/0.94	0.76/1.08	0.72/0.96	0.77/1.11	0.75/0.98	**0.58/0.88**
3.0s	<u>0.94/1.34</u>	1.06/1.40	0.98/1.35	1.06/1.43	1.00/1.39	1.11/1.46	1.03/1.41	**0.87/1.31**
4.0s	<u>1.17/1.61</u>	1.32/1.74	1.18/1.63	1.35/1.78	1.19/1.66	1.37/1.79	1.23/1.67	**1.13/1.60**
Acc.	0.69	0.65	<u>0.70</u>	0.68	0.69	0.64	0.65	**0.73**

Table 2: Quantitative results (ADE_{20}/FDE_{20}, and accuracy) of trajectory prediction in NBA dataset. The bold/underlined font represent the best/second best result.

conditions, including basketball and football games. But our logic-learning method tries to use a set of spatial-temporal logic rules with intention variables involved as principles to model the dynamics of human actions, not restrained into specific conditions.

Further, The proposed model achieved the best scores in F1 score, which is an balanced metric considering both recall and precision. This is because the rule generator and evaluator can collaborate with each other to reduce search space and learn better rules. More experiments and ablation studies can be found in Supplementary Material.

5.4 Visualization

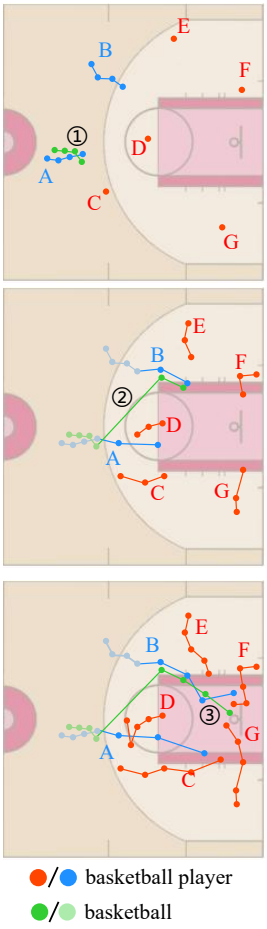
As mentioned before, pedestrian trajectory prediction is a complex problem because we have to consider the spatial-temporal properties of each pedestrian in the scene. Pedestrians in crowded scenes may have complex interactions, representing different motion modes, including forming groups, following other pedestrians, changing directions to avoid collisions, etc. We show some qualitative results for trajectory prediction on SDD dataset in Figure 3. Note that the second column has illuminate the most likely predicted goal in the following 30 seconds, where the higher color means higher probability. We mark some sampled goals in the orange cross. And the last column shows our predicted trajectories compared with ground truth. We observe that Y-net predicts diverse scene-complaint trajectories, with both future goals and paths modalities. The predicted trajectories are closer to socially-acceptable trajectories and forms more stable behaviors between members.

5.5 Estimated Intention

We provide a qualitative evaluation of the estimated distributions for for each player's intention in NBA dataset in Figure 4, representing spatial-temporal relations, which can be estimated from examples and used to manipulate a scene in order to fulfill spatial relations specified in verbal commands. Yellow regions have higher value, while purple regions have value close to zero. For the relations left of, right of, in front of, behind and other side of it is visible how entities exactly and roughly change the angle variance. Interestingly, for these intentions, the distributions seem to complement the affirmative exemplars mainly in terms of direction and distance. As a consequence, these distributions still represent locations in the area around the reference object.

5.6 Generated Logic Rules

We add visualization and explanation about the logic rule and corresponding actions from NBA dataset in Figure 5. Note that the static spatial relation $\{\texttt{Left(B,A)}\}$ represents that the player `B` is on the left of player `A`. And the dynamic spatial relation $\{\texttt{Away(A,E)}\}$ means that player `A` is getting away from player `E`. We can see that these logic rules are meaningful and diverse. In this picture, player A is defended by two players and then passes the ball to player B. Player B goes front and crossover to bypass three defenders and shoot at the basket. Our rule can actually represent their



Player A Rule ①: Left(B,A)∧Right(C,A)∧Front(D,A)→Pass(A,B)
Explanation:

- Player A is carrying the ball
- Player A is defended by player C and player D
- The partner B is on the left of player A
- Player A passes the ball to player B

Player B Rule ②: Left(E,B)∧Right(D,B)→Go_Front(B)
Explanation:

- Player E and player F is moving to the player B
- Player B is defended by player E on left side
- Player B is defended by player D on right side
- Player B goes front and carry the ball

Player B Rule ③: Left(E,B)∧Right(G,B)∧Front(F,B)→Shoot(B)
Explanation:

- Player B is defended by player E on left side
- Player B is defended by player G on right side
- Player B is defended by player F on front side
- Player B shoots at the basket

Figure 5: Visualization and explanation of logic rules in NBA dataset.

offensive strategy. In fact, our framework can actually adapt to some complex motions, such as cutting toward the ball, because these spatial-temporal predicates are fed into the rule generator and evaluator to obtain high-quality rules to explain the intention of players.

6 Limitation

It's challenging to define some complex predicates with richer meanings for different datasets. Given a more informative dataset, our method can discover more principles-like complex rules. Our motivation is to consider the spatial-temporal relation between pedestrians and generate some high-quality logic rules to explain their behaviors. Although we only choose some simple actions in our experiments, they can bring some benefits for understanding the principles of biological agents' behaviors. Our framework is suitable for more complex conditions, supposing more sophisticated action predicates can be obtained from the data.

7 Conclusion

We proposed a framework for learning intrinsic spatial-temporal logic rules for explaining human actions. We regard logic rules as latent variables, and the rule generator as well as the rule evaluator are jointly learned with EM-based algorithm. In the experiments, our method can analyze the biological movement sequence of pedestrians and players, and obtained novel insights from generated logic rules. In the future, we plan to incorporate other physical laws into the models, such as conservation of energy and momentum to enhance robustness of our model.

References

[1] David Adam. Science and the world cup: how big data is transforming football. *Nature*, 611(7936):444–446, 2022.

[2] Claudine Badue, Rânik Guidolini, Raphael Vivacqua Carneiro, Pedro Azevedo, Vinicius B Cardoso, Avelino Forechi, Luan Jesus, Rodrigo Berriel, Thiago M Paixao, Filipe Mutz, et al. Self-driving cars: A survey. *Expert Systems with Applications*, 165:113816, 2021.

[3] Ricky TQ Chen, Brandon Amos, and Maximilian Nickel. Neural spatio-temporal point processes. In *International Conference on Learning Representations*, 2021.

[4] Chang Choi, Junho Choi, Eunji Lee, Ilsun You, and Pankoo Kim. Probabilistic spatio-temporal inference for motion event understanding. *Neurocomputing*, 122:24–32, 2013.

[5] Sanjeeb Dash, Oktay Gunluk, and Dennis Wei. Boolean decision rules via column generation. *Advances in neural information processing systems*, 31, 2018.

[6] Tianpei Gu, Guangyi Chen, Junlong Li, Chunze Lin, Yongming Rao, Jie Zhou, and Jiwen Lu. Stochastic trajectory prediction via motion indeterminacy diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17113–17122, 2022.

[7] Yue Hu, Siheng Chen, Ya Zhang, and Xiao Gu. Collaborative motion prediction via neural motion message passing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6319–6328, 2020.

[8] Yingfan Huang, Huikun Bi, Zhaoxin Li, Tianlu Mao, and Zhaoqi Wang. Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6272–6281, 2019.

[9] Rainer Kartmann, You Zhou, Danqing Liu, Fabian Paus, and Tamim Asfour. Representing spatial object relations as parametric polar distribution for scene manipulation based on verbal commands. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8373–8380. IEEE, 2020.

[10] 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*, 2022.

[11] Karttikeya Mangalam, Yang An, Harshayu Girase, and Jitendra Malik. From goals, waypoints & paths to long term human trajectory forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15233–15242, 2021.

[12] Karttikeya Mangalam, Harshayu Girase, Shreyas Agarwal, Kuan-Hui Lee, Ehsan Adeli, Jitendra Malik, and Adrien Gaidon. It is not the journey but the destination: Endpoint conditioned trajectory prediction. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 759–776. Springer, 2020.

[13] Andrew Miller, Luke Bornn, Ryan Adams, and Kirk Goldsberry. Factorized point process intensities: A spatial analysis of professional basketball. In *International conference on machine learning*, pages 235–243. PMLR, 2014.

[14] Filmer Stuart Cuckow Northrop. The logic of the sciences and the humanities. 1947.

[15] Clayton Peterson. A logic for human actions. *Applications of Formal Philosophy: The Road Less Travelled*, pages 73–112, 2017.

[16] Meng Qu, Junkun Chen, Louis-Pascal Xhonneux, Yoshua Bengio, and Jian Tang. Rnnlogic: Learning logic rules for reasoning on knowledge graphs. In *International Conference on Learning Representations*, 2021.

[17] Alex Reinhart. A review of self-exciting spatio-temporal point processes and their applications. *Statistical Science*, 33(3):299–318, 2018.

[18] Davis Rempe, Tolga Birdal, Yongheng Zhao, Zan Gojcic, Srinath Sridhar, and Leonidas J Guibas. Caspr: Learning canonical spatiotemporal point cloud representations. *Advances in neural information processing systems*, 33:13688–13701, 2020.

[19] Ali Sadeghian, Mohammadreza Armandpour, Patrick Ding, and Daisy Zhe Wang. Drum: End-to-end differentiable rule mining on knowledge graphs. *Advances in Neural Information Processing Systems*, 32, 2019.

[20] Bilong Shen, Xiaodan Liang, Yufeng Ouyang, Miaofeng Liu, Weimin Zheng, and Kathleen M Carley. Stepdeep: A novel spatial-temporal mobility event prediction framework based on deep neural network. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 724–733, 2018.

[21] Li-Wu Tsao, Yan-Kai Wang, Hao-Siang Lin, Hong-Han Shuai, Lai-Kuan Wong, and Wen-Huang Cheng. Social-ssl: Self-supervised cross-sequence representation learning based on transformers for multi-agent trajectory prediction. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII*, pages 234–250. Springer, 2022.

[22] Tong Wang, Cynthia Rudin, Finale Doshi-Velez, Yimin Liu, Erica Klampfl, and Perry MacNeille. A bayesian framework for learning rule sets for interpretable classification. *The Journal of Machine Learning Research*, 18(1):2357–2393, 2017.

[23] Chenxin Xu, Maosen Li, Zhenyang Ni, Ya Zhang, and Siheng Chen. Groupnet: Multiscale hypergraph neural networks for trajectory prediction with relational reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6498–6507, 2022.

[24] Fan Yang, Zhilin Yang, and William W Cohen. Differentiable learning of logical rules for knowledge base reasoning. *Advances in neural information processing systems*, 30, 2017.

[25] Cunjun Yu, Xiao Ma, Jiawei Ren, Haiyu Zhao, and Shuai Yi. Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*, pages 507–523. Springer, 2020.

[26] Jiangbei Yue, Dinesh Manocha, and He Wang. Human trajectory prediction via neural social physics. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIV*, pages 376–394. Springer, 2022.

[27] Zihao Zhou, Xingyi Yang, Ryan Rossi, Handong Zhao, and Rose Yu. Neural point process for learning spatiotemporal event dynamics. In *Learning for Dynamics and Control Conference*, pages 777–789. PMLR, 2022.