Census Signal Temporal Logic Inference for Multiagent Group Behavior Analysis

Zhe Xu, *Student Member, IEEE*, and A. Agung Julius, *Member, IEEE*

***Abstract*— In this paper, we define a novel census signal temporal logic (CensusSTL) that focuses on the number of agents in different subsets of a group that complete a certain task specified by the STL. CensusSTL consists of an "*inner logic*" STL formula and an "*outer logic*" STL formula. We present a new inference algorithm to infer CensusSTL formulas from the trajectory data of a group of agents. We first identify the "*inner logic*" STL formula and then infer the subgroups based on whether the agents' behaviors satisfy the "*inner logic*" formula at each time point. We use two different approaches to infer the subgroups based on similarity and complementarity, respectively. The "*outer logic*" CensusSTL formula is inferred from the census trajectories of different subgroups. We apply the algorithm in analyzing data from a soccer match by inferring the CensusSTL formula for different subgroups of a soccer team.**

***Note to Practitioners*—The method described in this paper can be used to discover subgroups of agents and their behavior patterns based on spatial–temporal data of all agents in the group. The subgroups are formed by agents that behave similarly or complementarily for a certain task. The behavior patterns of the subgroups are expressed in the form of signal temporal logic statements. These are logical statements about the number of agents in the subgroups that are in certain states during a certain time interval. For example, if there are more than two blue agents in the region, then within 2 h, there will be more than three red agents in the same region. The behavior patterns inferred using this method can be used in: 1) analysis of group behavior while performing a task, e.g., in sports tactical analysis and 2) prediction of future group behavior under similar situations.**

***Index Terms*— Census signal temporal logic (CensusSTL), group behavior, multiagent systems.**

I. Introduction

IN SOME multiagent systems, there are subgroups that perform different tasks, such as the defenders, midfielders, and forward in a soccer team [1]. Within each subgroup, the agents can be seen as interchangeable members in the sense that as long as there is a certain number of agents in the subgroup performing the task, it does not matter who these agents are. In the social network context, the behavior pattern of different groups of people and the temporal influence on them have been a research focus [2]–[4]. A recommender system can use this information to give better recommendations of the place and time for doing certain activities whether it is shopping or checking in at a hotel. In robotics, the multiagent robot systems (MARS) [5]–[8] are being studied for their co-operative behaviors, such as the leader robot, tracking a prescribed trajectory, and the rest of the robots, following the leader while forming a desired formation pattern [9]. In all of these applications, how to express and characterize the properties of the group behavior has always been a challenge.

A. Related Works

There has been rich literature on the formalization of multiagent group behaviors. Wang *et al.* [10] propose an ontology-based behavior modeling and checking system to explicitly represent and verify complex group behavior interactions. Temporal logic is a formal approach that has been increasingly used in expressing more complicated and precise high-level control specifications [11], [12]. There has been many different temporal logic frameworks in multiagent systems to guarantee safe and satisfactory performance from high-level perspectives, such as linear temporal logic (LTL) [13], [14], computational tree logic (CTL) [15], alternating-time temporal logic (ATL) [16], etc. The temporal logic formulas are predefined as a specification for the behaviors of the system [17], [18].

Recently, there is a growing interest in devising algorithms to identify dense-time temporal logic formulas from system trajectories [19]. Asarin *et al.* [20] present a method to synthesize magnitude and timing parameters in a quantitative temporal logic formula, so that it fits observed data. Kong *et al.* [21] designed an inference algorithm that can automatically construct signal temporal logic (STL) formulas directly from data. The obtained STL formulas can be used to classify different behaviors, predict future behaviors, and detect anomaly behaviors [22].

B. Contributions and Advantages

In this paper, we define a novel census STL (CensusSTL) that focuses on the number of agents and the structure of the group that complete a certain task specified by the STL. The word "*census* " means "*the procedure of systematically acquiring and recording information about the members of a given population*" [23]. In the group behavior analysis,

Manuscript received March 23, 2016; revised August 10, 2016; accepted September 11, 2016. Date of publication October 10, 2016; date of current version January 4, 2018. This paper was recommended for publication by Associate Editor D. V. Dimarogonas and Editor D. Han upon evaluation of the reviewers' comments. This work was supported by the National Science Foundation under Grant CNS-0953976 and Grant CNS-1218109.

The authors are with the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180 USA (e-mail: xuz8@rpi.edu; juliua2@rpi.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASE.2016.2611536

we need to generate knowledge about the behaviors of the members or agents of different subgroups, and CensusSTL provides a formal structure for generating such knowledge. The CensusSTL formula is essentially an STL formula ("*outer logic*") with the variable in the predicate being the number of agents whose behaviors satisfy another STL formula ("*inner logic*"). For example, the CensusSTL formula can express specifications, such as "*from 10* A.M. *to 2* P.M.*, at least three policemen should be present at the lobby for at least 20 min in every hour*," where the "*inner logic*" formula is the task "*be present at the lobby for at least 20 min in every hour*" and the "*outer logic*" formula is "*from 10* A.M. *to 2* P.M.*, and at least three policemen should perform the task*."

CensusSTL is different from the other temporal logic frameworks for multiagent systems as it does not focus on individual agents or the interaction between different agents, but on the number of agents in different subgroups that complete a certain task. Therefore, it is more useful in applications where only the number of agents or the proportion of agents in different subgroups of a population is of interest while different agents in a subgroup can be seen as interchangeable.

We present a new inference algorithm that can infer the CensusSTL formula directly from individual agent trajectories. Our inference method for the "*inner logic*" formula and the "*outer logic*" formula is similar to [20] as we also choose the template of formula first and then search for the parameters. However, we formulate the problem as a group behavior analysis problem, so the objective of our approach is not only finding the parameters that fit certain temporal logic formula, but also infer the subgroups and the temporal relationship among the different subgroups.

C. Organizations

This paper is structured as follows. Section II introduces the framework of CensusSTL. Section III shows the algorithm to infer the census temporal logic formula from data. Section IV implements the algorithm on analyzing a soccer match as a case study. Finally, some conclusions are presented in Section V.

II. Census Signal Temporal Logic

A. "Inner Logic" Signal Temporal Logic

In this paper, we find subgroups of a population that act collaboratively for a task. We need to find both the task and the subgroups from the time-stamped trajectories (for the definition of time-stamped trajectories, see the beginning of Section II-B) of different agents. The task can be formulated as an "*inner logic*" STL formula. Assume there is a group S of n agents and each agent has an observation space $\mathbb{X}$. For example, the group can be a set of points moving in 2-D plane, and the observation space can be their 2-D positions. Each element of the observation space is described by a set of w variables that can be written as a vector $x = [x_1, x_2, \ldots, x_w]^T$. The domain of x is denoted by $\mathbb{X} = \mathbb{X}_1 \times \mathbb{X}_2 \times \cdots \times \mathbb{X}_w$. The domain $\mathbb{B}$ ={true, false} is the Boolean domain and the time set is $\mathbb{T} = \mathbb{R}$ (note that we allow negative time to add more flexibility of the temporal operator). With a slight abuse of notation, we define observation trajectory (or signal or behavior) x describing the observation of each agent as a function from $\mathbb{T}$ to $\mathbb{X}$. Therefore, x_i refers to both the name of the ith observation variable and its valuation in $\mathbb{X}$. A finite set $M = \{\mu_1, \mu_2, \ldots \mu_q\}$ is a set of atomic predicates, each mapping $\mathbb{X}$ to $\mathbb{B}$. The "*inner logic*" is STL [24] and the syntax of the "*inner logic*" STL proposition can be defined recursively as follows:

$$\phi := \top \mid \mu \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \phi_1 \mathcal{U}_I \phi_2$$

where $\top$ stands for the Boolean constant true, μ is an atomic predicate in the form of an inequality $f(x(t)) > 0$ where f is some real-valued function, $\neg$ (negation), $\wedge$ (conjunction), and $\vee$ (disjunction) are standard Boolean connectives, $\mathcal{U}$ is a temporal operator representing "until," and I is an interval of the form $I = (i_1, i_2), (i_1, i_2], [i_1, i_2)$ or $[i_1, i_2]$ $(i_1 \leq i_2, i_1, i_2 \in \mathbb{T})$. In general, a predicate can be an atomic predicate or atomic predicates connected with standard Boolean connectives. We can also derive two useful temporal operators from "until" $(\mathcal{U})$, which are "eventually" $\Diamond\phi = \top\mathcal{U}\phi$ and "always" $\Box\phi = \neg\Diamond\neg\phi$.

We use (x, t) to represent the observation trajectory x at time t; then, the Boolean semantics of "*inner logic*" are defined recursively as follows:

$$\begin{aligned}
&(x,t) \models \mu \quad \text{iff } f(x(t)) > 0\\
&(x,t) \models \neg\phi \quad \text{iff } (x,t) \not\models \phi\\
&(x,t) \models \phi_1 \wedge \phi_2 \quad \text{iff } (x,t) \models \phi_1 \text{ and } (x,t) \models \phi_2\\
&(x,t) \models \phi_1 \vee \phi_2 \quad \text{iff } (x,t) \models \phi_1 \text{ or } (x,t) \models \phi_2\\
&(x,t) \models \phi_1 \mathcal{U}_{[a,b)} \phi_2 \quad \text{iff } \exists t' \in [t+a, t+b)\\
&\qquad \text{s.t. } (x,t') \models \phi_2, \quad (x,t'') \models \phi_1\\
&\qquad \forall t'' \in [t+a, t').
\end{aligned}$$

The robustness degree of an observation trajectory x with respect to an "*inner logic*" formula ϕ at time t is given as $r(x, \phi, t)$, where r can be calculated recursively via the quantitative semantics [24]

$$\begin{aligned}
r(x,\mu,t) &= f(x(t))\\
r(x,\neg\phi,t) &= -(r(x,\phi,t))\\
r(x,\phi_1\wedge\phi_2,t) &= \min(r(x,\phi_1,t), r(x,\phi_2,t))\\
r(x,\phi_1\vee\phi_2,t) &= \max(r(x,\phi_1,t), r(x,\phi_2,t))\\
r(x,\Box_{[\tau_1,\tau_2)}\phi,t) &= \min_{t+\tau_1\leq t'<t+\tau_2} r(x,\phi,t')\\
r(x,\Diamond_{[\tau_1,\tau_2)}\phi,t) &= \max_{t+\tau_1\leq t'<t+\tau_2} r(x,\phi,t')\\
r(x,\phi_1\mathcal{U}_{[a,b)}\phi_2,t) &= \sup_{t+a\leq t'<t+b} \Big(\min\Big(r(s,\phi_2,t'),\\
&\qquad \sup_{t\leq t''<t'} r(s,\phi_1,t'')\Big)\Big).
\end{aligned}$$

B. Signal Temporal Logic Applied to Data

We make two deviations to STL when applying an "*inner logic*" formula ϕ to data.

1) As the observation trajectory is usually of finite length, and also considering that there may be negative time in the temporal operator of the "*inner logic*" formula ϕ, the satisfaction of the "*inner logic*" formula ϕ

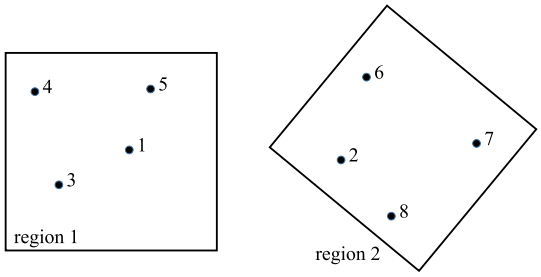

Fig. 1. Eight agents and two regions in the furniture moving example.

may not be well defined at every time point of the observation trajectory (for example, for the formula $\phi_1 = \Diamond_{[0,10)}(x > 5)$ and $\phi_2 = \Diamond_{[-10,0)}(x > 5)$, if the observation trajectory is defined on the time domain of [0, 200], then ϕ_1 can only be evaluated on the time domain of [0, 190] and ϕ_2 can only be evaluated on the time domain of [10, 200]). Assume that the time domain of the observation trajectory x is $\mathbb{T}_o \subset \mathbb{T}$, with a slight abuse of notation, we define time-stamped trajectory x of finite length as a function from $\mathbb{T}_o$ to $\mathbb{X}$.

The time domain of the "*inner logic*" formula ϕ with respect to x is defined recursively as follows:

$$\begin{aligned} D(\mu, x) &= \mathbb{T}_o \\ D(\neg\phi, x) &= D(\phi, x) \\ D(\phi_1 \wedge \phi_2, x) &= D(\phi_1, x) \cap D(\phi_2, x) \\ D(\phi_1 \mathcal{U}_{[a,b)} \phi_2, x) &= \{t \mid [t+a, t+b) \\ &\quad \subset (D(\phi_1, x) \cap D(\phi_2, x))\}. \end{aligned}$$

For example, for "*inner logic*" formula $\phi = \Diamond_{[0,10)}(x > 5) \wedge \Diamond_{[20,40)}(\Box_{[20,60)}(x > 20))$, if the observation trajectory is defined on $[0, 200]$, then $D(\phi) = [0-0, 200-10] \cap [0-20-20, 200-40-60] = [0, 100]$.

2) The observation data are usually discrete, so the time domain of the observation trajectory $\mathbb{T}_o$ is a set of discrete-time points. In this case, the interval I in the form $I = (i_1, i_2), (i_1, i_2], [i_1, i_2)$ or $[i_1, i_2]$ actually means the time points in $\mathbb{T}_o$ that belongs to I. For example, $[i_1, i_2)$ is interpreted as $\{t \in \mathbb{T}_o \mid t \in [i_1, i_2)\}$.

Consider the example shown in Fig. 1 where there are two predicates p_{region1} and p_{region2} corresponding to Region 1 and Region 2 [for the representation of predicates, see (4) in Section III] and eight different agents (people) who are moving furniture from Region 1 to Region 2. The people need to move back and forth frequently between Region 1 and Region 2. One STL formula that can characterize the people moving pattern is

$$\phi_t = \Box_{[0,\tau_1)} p_{\text{region1}} \wedge \Diamond_{[\tau_2,\tau_3)}(p_{\text{region2}} \wedge \Diamond_{[\tau_4,\tau_5)} p_{\text{region1}}) \quad (1)$$

which reads that "*the person is in Region 1 for τ_1 time units and arrives in Region 2 sometime between τ_2 and τ_3 time units, then sometime between τ_4 and τ_5 time units later the person comes back to Region 1.*" The temporal parameters satisfy $\tau_5 \geq \tau_4$, $\tau_3 \geq \tau_2$, and $\tau_1 \geq 0$.

We specify that the "*inner logic*" formula $\phi = \Diamond_{[-\tau_5-\tau_3,0]}\phi_t$ while the temporal operator $\Diamond_{[-\tau_5-\tau_3,0]}$ is to make ϕ true at every time point during the execution of the task. Without this temporal operator, ϕ is only true at the beginning of the task.

C. "Outer Logic" Census Signal Temporal Logic

Based on the "*inner logic*," we can define the "*outer logic*" CensusSTL. The observation element of the "*outer logic*" is the number of agents that satisfy the "*inner logic*" formula, which can be described by nonnegative integers that belong to the domain $\mathbb{N}$. A census trajectory $n(\phi, S)$ describes the number of agents in the group S whose behaviors satisfy the "*inner logic*" formula ϕ over time.

It should be noted that the time domain of the census trajectories is the same as the time domain of the "*inner logic*" formula ϕ with respect to observation trajectory x if the observation trajectory is of finite length, so $n(\phi, S)$ is a mapping from $D(\phi, x)$ to $\mathbb{N}$. As there may be different subgroups $S_i (i = 1, 2, \ldots)$ in group S, we have Definition 1.

Definition 1: We define $n(\phi, S_i, t)$ as the number of agents in the subgroup $S_i (i = 1, 2, \ldots)$ whose behaviors satisfy the "*inner logic*" formula ϕ at time t, or in other words, the number of agents whose behavior (observation trajectory) has positive robustness degree with respect to ϕ at time t.

With that notation, the atomic predicate of the "*outer logic*" CensusSTL can be defined as follows:

$$\mu_n := n(\phi, S_i) > c \mid -n(\phi, S_i) > -c \quad (2)$$

where c is a nonnegative integer.

In the furniture moving example, assume that there are two subgroups of people who are moving the furniture, $\{1, 2, 3, 4\}$ and $\{5, 6, 7, 8\}$. The atomic predicate of the "*outer logic*" can express properties, such as "*the number of people in the subgroup $\{1, 2, 3, 4\}$ who are moving the furniture is less than 2*," or "*the number of people in the subgroup $\{5, 6, 7, 8\}$ who are moving the furniture is more than 3.*"

The syntax of the "*outer logic*" CensusSTL proposition can be defined recursively as follows:

$$\gamma := \top \mid \mu_n \mid \neg\gamma \mid \gamma_1 \wedge \gamma_2 \mid \gamma_1 \vee \gamma_2 \mid \gamma_1 \mathcal{U}_I \gamma_2. \quad (3)$$

As the "*outer logic*" CensusSTL is also STL, the semantics of STL also applies to the "*outer logic*" CensusSTL. The robustness degree of a census trajectory n with respect to a CensusSTL formula γ at time t is denoted as $r(n, \gamma, t)$, where r can be calculated recursively in the same way as $r(x, \phi, t)$ is calculated.

III. Census Signal Temporal Logic Inference

In this section, we seek to infer the CensusSTL formula describing the behaviors of a group of agents from the collection of the individual agent observation trajectories in a training data set and then test the validity of the inferred CensusSTL formula in a separate validation data set. We choose to represent the predicates as polyhedral sets as they are more general than rectangular sets and computationally easier to handle than other more complex sets (ellipsoidal sets, nonconvex sets, etc.). So each predicate in the "*inner logic*"

formula is represented in the following form:

$$p := \left(\bigwedge_{k=1}^{m} a_k^T x > b_k \right), \quad a_k \in \mathbb{R}^n, \ b_k \in \mathbb{R} \tag{4}$$

where vector a_k and number b_k denote the parameters that define the predicate and m is the number of atomic predicates in the predicate.

According to the quantitative semantics of STL, the robustness of each predicate p can be expressed as the minimum of robustness of each atomic predicate

$$r(x, p, t) = \min_{1 \le k \le m} \left(a_k^T x(t) - b_k \right), \quad a_k \in \mathbb{R}^n, \ b_k \in \mathbb{R}. \tag{5}$$

A. Task Description

In this paper, we infer the "*inner logic*" STL formula in the form of $\phi = \Diamond_{[-\|\phi_t\|,0]}\phi_t$, where ϕ_t is the formula that describes the task with all the temporal parameters of ϕ_t chosen in $\mathbb{R}_{\geqslant 0}$ and $\|\phi_t\|$ is the necessary length associated with formula ϕ_t as defined in the following:

$$\|\phi_t\| := \min\{T \mid \text{if } \mathbb{T}_o = [0, T], D(\phi_t, x) \neq \emptyset\}.$$

Take STL formula $\phi_t = \Diamond_{[0,10)}(x > 5) \wedge \Diamond_{[20,40)}(\Box_{[20,60)}(x > 20))$ for example, the necessary length $\|\phi_t\| = 40 + 60 = 100$ and $\phi = \Diamond_{[-100,0]}\phi_t$ is true at every time point during the execution of the task. We consider four templates of temporal logic formula ϕ_t corresponding to four common tasks.

1) Sequential Task:

$$\begin{aligned} \phi_t = {} & \Box_{[0,\tau_1)}\phi_{t1} \wedge \Diamond_{[\tau_{21},\tau_{22})}\Box_{[0,\tau_{23})}\phi_{t2} \\ & \wedge \cdots \wedge \Diamond_{[\tau_{z1},\tau_{z2})}\Box_{[0,\tau_{z3})}\phi_{tz} \end{aligned} \tag{6}$$

where $\phi_{t1}, \phi_{t2}, \ldots, \phi_{tz}$ are subtasks that can be predicates or STL formulas as ϕ_t and the temporal parameters satisfy $\tau_{21} \geq \tau_1$, $\tau_{i2} + \tau_{i3} \leq \tau_{j1} (2 \leq i < j \leq z)$. For any term $\Diamond_{[\tau_{i1},\tau_{i2})}\Box_{[0,\tau_{i3})}\phi_{ti}$ $(i = 2, 3, \ldots, z)$, if $\tau_{i1} = \tau_{i2}$, then this term shrinks to $\Box_{[\tau_{i1},\tau_{i1}+\tau_{i3})}\phi_{ti}$; if $\tau_{i3} = 0$, then this term shrinks to $\Diamond_{[\tau_{i1},\tau_{i2})}\phi_{ti}$. The sequential task is a series of subtasks that are performed in a sequential order.

2) Concurrent Task:

$$\phi_t = \Box_{[0,\tau_1)}(\phi_{t1} \vee \phi_{t2} \ldots \vee \phi_{tz}). \tag{7}$$

This concurrent task means "*during the next τ_1 time units, the agent performs at least one of the subtasks ϕ_{ti}.*"

3) Persistent Task:

$$\phi_t = \Box_{[0,\tau_1)}\Diamond_{[0,\tau_2)}\phi_{t1}. \tag{8}$$

This persistent task means "*during the next τ_1 time units, ϕ_{t1} is performed at least once in every τ_2 time units.*"

4) Causal Task:

$$\phi_t = \Box_{[0,\tau_1)}(\phi_{t1} \Rightarrow \phi_{t2}) \tag{9}$$

where ϕ_{t1} is the cause formula and ϕ_{t2} is the effect formula. This causal task means "*during the next τ_1 time units, whenever the subtask ϕ_{t1} is performed, the agent will perform subtask ϕ_{t2}.*"

In all of these task templates, we set an upper limit to the necessary length associated with formula ϕ_t as we only consider tasks that are finished within a certain time. For example, if it generally takes no more than ten time units to move the furniture from Region 1 to Region 2, then we set $\|\phi_t\| \leq 10$.

In the following, we introduce the specific steps to infer the CensusSTL formula from data. Note that our procedure cannot produce a formula that does not conform with the predetermined templates. Our aim is to find the CensusSTL formula that best fits (according to some measure of fitness) a given finite set of observation trajectories. In general, we are given a training data set of z different observation trajectories for each agent, where the time domains of the z observation trajectories are not necessarily the same.

B. "Inner Logic" STL Formula Inference

In this section, we discuss the three requirements the "*inner logic*" formula needs to meet and then formulate the optimization problem for the "*inner logic*" formula inference.

1) Consistency: We heuristically postulate that if the number of agents whose behaviors satisfy the "*inner logic*" formula is changing drastically through time, then the formula cannot reflect a task that a group of agents are performing consistently.

Definition 2: We define $v_q(\phi, S)$ as the temporal variation of the number of agents in the set S whose qth observation trajectories satisfy the "*inner logic*" formula ϕ, which can be described as follows:

$$v_q(\phi, S) = \frac{1}{l_{\phi,q} - 1} \sum_{j=1}^{l_{\phi,q}-1} |n_q(\phi, S, j+1) - n_q(\phi, S, j)| \tag{10}$$

where $n_q(\phi, S, j)$ is the number of agents in the set S whose qth observation trajectories satisfy the STL formula ϕ at the jth time point, and $l_{\phi,q}$ is the number of time points in the time domain of the qth census trajectory.

2) Frequency: We postulate that if the number of time points at which the behavior of any agent satisfies the "*inner logic*" formula ϕ is small, then the formula cannot reflect a task that is performed frequently.

Definition 3: We define $m(\phi, k)$ as the total number of time points at which the "*inner logic*" formula ϕ is true for agent k in the training data set (the time points in different observation trajectories are counted separately).

3) Specificity: Sometimes, a consistent and frequent task can be overly general or meaningless. For example, the proposition "*the agent is always in the entire space $\mathbb{X}$*" is always true but does not contain any useful information. To make the task more specific and meaningful, we incorporate some *a priori* knowledge about the system. The other purpose of incorporating *a priori* knowledge is to make the task more tailored to the user preferences. For example, if the user is particularly interested in the behavior in a certain region, then this region can be specified as an *a priori* predicate. Suppose that we are given *a priori* predicates $X_i (i = 1, 2, \ldots, n_p)$, we make the obtained predicates p_i as similar as possible to the *a priori* predicates X_i. The Hausdorff distance is an important tool to measure the similarity between two sets

of points [25]. It is defined as the largest distance from any point in one of the sets, to the closest point in the other set. Suppose that the set of states that satisfy the predicate p is $\mathcal{O}(p) \subset \mathbb{X}$. Then, the Hausdorff distance $d_H(\mathcal{O}(X_i), \mathcal{O}(p_i))$ is expressed as follows:

$$d_H(\mathcal{O}(X_i), \mathcal{O}(p_i)) = \max\Big\{ \sup_{x\in\mathcal{O}(X_i)} \inf_{y\in\mathcal{O}(p_i)} d(x,y), \sup_{y\in\mathcal{O}(p_i)} \inf_{x\in\mathcal{O}(X_i)} d(x,y)\Big\}. \quad (11)$$

The expression $\sup_{x\in\mathcal{O}(X_i)} \inf_{y\in\mathcal{O}(p_i)} d(x,y)$ when both $\mathcal{O}(X_i)$ and $\mathcal{O}(p_i)$ are convex polyhedra can be evaluated as follows.

Step 1: Calculate all vertices of the polyhedron $\mathcal{O}(X_i)$. Denote them as $\psi_1, \psi_2, \ldots, \psi_{N_{\mathcal{O}(X_i)}}$.

Step 2: Calculate the distance from ψ_j to $\mathcal{O}(p_i)$ for each $j \in \{1, \ldots, N_{\mathcal{O}(X_i)}\}$. This is a convex quadratic optimization problem.

Step 3: Find the maximum of the distances calculated in Step 2.

We denote all parameters that define the "*inner logic*" STL formula ϕ as α. Take the case of $\phi = \square_{[\tau_1,\tau_2)}(\bigwedge_{k=1}^{m} a_k^T x > b_k)$, for example. As $x + 2y > 4$ and $2x+4y > 8$ are essentially the same, we constraint $\|a_k\|_2$ to be 1. One simple way to remove this constraint is to represent a_k using trigonometric parameters $\theta_{k,1}, \theta_{k,2}, \ldots$ Then, $a_k^T x$ can be represented as $\cos(\theta_{k,1})x_1 + \sin(\theta_{k,1})\cos(\theta_{k,2})x_2 + \sin(\theta_{k,1})\sin(\theta_{k,2})x_3 + \cdots$ by utilizing the fact that $\sin^2(\theta_{k,j}) + \cos^2(\theta_{k,j}) = 1$. For the formula $\phi(\alpha)$ mentioned earlier, $\tau_1, \tau_2, \theta_{k,j}$, and b_k are the elements of α. The lower bound and upper bound of the angles are set to be $[-\pi, \pi]$.

To summarize the three requirements, the inference of the "*inner logic*" formula $\phi = \Diamond_{[-\|\phi_t\|,0]}\phi_t$ where ϕ_t conforms to 1 of the four task templates is a constrained multiobjective problem, that is

Objectives:

$$\begin{aligned}
&\min \sum_{q=1}^{z} v_q(\phi(\alpha), S) \ \text{(consistency)}\\
&\max \sum_{k=1}^{n} m(\phi(\alpha), k) \ \text{(frequency)}\\
&\min \sum_{i=1}^{n_p} d_H(\mathcal{O}(X_i), \mathcal{O}(p_i(\alpha))) \ \text{(specificity)}\\
&\text{s.t.: } \|\phi_t(\alpha)\| \le \tau_{\text{limit}}
\end{aligned}$$

where α is the optimization variable and τ_{limit} is the upper limit of the necessary length associated with formula $\phi_t(\alpha)$.

We use particle swarm optimization [26] to optimize α (including the spatial parameters θ, b and the temporal parameters τ) of each possible "*inner logic*" formula. In each iteration, the parameters are updated as a swarm of particles that move in the parameter space to find the global minimum (in this paper, we use 200 particles for each iteration). The formula with the smallest value of the cost function can be generated and selected. The cost function is as follows:

$$J_{\text{stl}}(\phi, \alpha) = \sum_{q=1}^{z} v_q(\phi(\alpha), S) - \lambda_1 \sum_{k=1}^{n} m(\phi(\alpha), k) + \lambda_2 \sum_{i=1}^{n_p} d_H(\mathcal{O}(X_i), \mathcal{O}(p_i(\alpha)) \quad (12)$$

where λ_1 and λ_2 are weighting factors that can adjust the priorities of the different optimization goals (for tuning of λ_1 and λ_2, see the example in Section IV).

C. Group Partition

As there may be subgroups in the group, we proceed to infer the subgroups based on the identified formula $\phi(\alpha^*)$ where α^* minimizes J_{stl}.

Definition 4: The signature $s_q(\phi(\alpha^*), k, t)$ is defined as the satisfaction signature of the agent k with respect to the "*inner logic*" formula $\phi(\alpha^*)$ at time t in the qth observation trajectory. If the agent k satisfies $\phi(\alpha^*)$ at time t in the qth observation trajectory, then the signature is set to 1 at that time point; otherwise, it is set to 0.

We need to cluster the agents of the group into subgroups based on the satisfaction signature trajectories of different agents. For a given set of n elements, the number of all possible partitions of the set where each partition has exactly n_c nonempty subsets is Stirling's number of the second kind [27]. The search over all possible partitions of a set is an NP-complete problem, and the calculation soon becomes intractable when the number of elements in the set increases. In order to reduce the calculation, we further look into two kinds of relationships: complementarity and similarity.

We come back to the furniture moving scenario and assume that there are two subgroups of people who are moving the furniture, $\{1, 2, 3, 4\}$ and $\{5, 6, 7, 8\}$.

Case 1: The two subgroups take turns to move the furniture from Region 1 to Region 2. For example, if the people in subgroup $\{1, 2, 3, 4\}$ move the furniture for 1 h, then the people in subgroup $\{5, 6, 7, 8\}$ will move the furniture for the next hour. Therefore, people in the same subgroup behave similarly.

Case 2: There are people from both the two subgroups who move the furniture from Region 1 to Region 2. For example, if there are always one person from subgroup $\{1, 2, 3, 4\}$ and two people from subgroup $\{5, 6, 7, 8\}$ who move the furniture for 1 h, then there will be always two people from subgroup $\{1, 2, 3, 4\}$ and one person from subgroup $\{5, 6, 7, 8\}$ who move the furniture for the next hour. In this case, people in the same subgroup behave complementarily in the sense that a certain number of people in the subgroup should perform the task.

Overall, both complementary and similar relationships can lead to interesting group behaviors, but with their different nature they should be dealt with differently.

1) Group Partition Based on Similarity: A lot of clustering methods are based on similarity. For example, k-means clustering is frequently used in partitioning n observations into k clusters, where each observation belongs to the cluster

TABLE I
$s(\phi(\alpha^*), k, t)$ FOR EIGHT AGENTS AND EIGHT TIME POINTS

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	$supp$ $(k, \phi(\alpha^*))$
Agent 1	1	1	0	0	1	1	0	0	0.5
Agent 2	1	1	0	0	1	1	0	0	0.5
Agent 3	1	1	0	0	1	1	0	0	0.5
Agent 4	0	0	1	1	0	0	1	1	0.5
Agent 5	0	0	1	1	0	0	1	1	0.5
Agent 6	0	0	1	1	0	0	1	1	0.5
Agent 7	1	0	0	0	0	0	0	0	0.125
Agent 8	0	0	0	1	0	0	0	0	0.125

with the nearest mean. However, its performance can be distorted when clustering high-dimensional data [28]. As we cluster different agents based on their satisfaction signatures at different time points (which is high-dimensional when dealing with lengthy time-series data), we need to use some other methods. One way is to represent the agents in the group as vertices of a weighted hypergraph (a hypergraph is an extension of a graph in the sense that each hyperedge can connect more than two vertices) and represent the relationship among different agents as hyperedges. Then, the clustering problem is transformed to a hypergraph-partitioning problem where a number of graph-partitioning software packages can be utilized. For example, hMETIS is a software package that can partition large hypergraphs in a fast and efficient way [29]. hMETIS can partition the vertices of a hypergraph, such that the number of hyperedges connecting vertices in different parts is minimized (minimal cut). The complexity of hMETIS for a k-way partitioning is $O((V + E) \log k)$ where V is the number of vertices and E is the number of edges [30]. In this paper, we modify the method in [30], which uses frequent item sets found by the association rule algorithm as hyperedges. *A priori* algorithm [31] is often used in finding association rules in data mining. It proceeds by identifying the frequent individual items[1] and extending them to larger and larger frequent item sets.[2] In this paper, we consider the different agents as items and an item "appears" whenever the satisfaction signature of the agent is 1.

Definition 5: We define the relative support $\text{supp}(k, \phi(\alpha^*))$ of agent k with respect to the "*inner logic*" formula $\phi(\alpha^*)$ as the proportion of satisfaction signatures of the agent k with respect to $\phi(\alpha^*)$ that are not zero, as shown in the following:

$$\text{supp}(k, \phi(\alpha^*)) \triangleq \frac{1}{\sum_{q=1}^{z} l_{\phi(\alpha^*),q}} \sum_{q=1}^{z} \sum_{j=1}^{l_{\phi(\alpha^*),q}} s_q(\phi(\alpha^*), k, j). \tag{13}$$

We give a simple example of eight agents and one observation trajectory of 8 time points (representing eight consecutive hours) for each agent in the furniture moving scenario with the signature $s(\phi(\alpha^*), k, t)$ listed in Table I. We first put all agents that are identified as frequent individual items in S_f, as shown in the following:

$$S_f \triangleq \{k \in S \mid \text{supp}(k, \phi(\alpha^*)) > \text{minsup}\} \tag{14}$$

where minsup is a small positive number as a threshold for defining frequent item sets. It can be seen from Table I that agent 7 and agent 8 do not perform the "*inner logic*" task as frequently as the other players, so we set minsup $= 0.2$ to exclude them from the partitioning process (in similarity relationships, all the agents in each subgroup are expected to perform the task frequently). In this case, $S_f = \{1, 2, 3, 4, 5, 6\}$.

Definition 6: The signature $s_q(\phi(\alpha^*), e, t)$ is defined as the satisfaction signature of the set e with respect to the "*inner logic*" formula $\phi(\alpha^*)$ at time t in the qth observation trajectory. If all the agents in the set e satisfy $\phi(\alpha^*)$ at time t in the qth observation trajectory, then the signature $s_q(\phi(\alpha^*), e, t)$ is set to 1 at that time point; otherwise, it is set to 0.

Definition 7: We denote the relative support $\text{supp}(e, \phi(\alpha^*))$ of a set e with respect to the "*inner logic*" formula $\phi(\alpha^*)$ as the proportion of satisfaction signatures of the set e with respect to $\phi(\alpha^*)$ that are not zero, as shown in the following:

$$\text{supp}(e, \phi(\alpha^*)) \triangleq \frac{1}{\sum_{q=1}^{z} l_{\phi(\alpha^*),q}} \sum_{q=1}^{z} \sum_{j=1}^{l_{\phi(\alpha^*),q}} s_q(\phi(\alpha^*), e, j). \tag{15}$$

If the relative support of a set e satisfies $\text{supp}(e, \phi(\alpha^*)) > \text{minsup}$, then we assign a hyperedge connecting the vertices (agents) of e, and the weight of hyperedge e is defined as relative support $\text{supp}(e, \phi(\alpha^*))$

$$\text{Weight}(e) = \text{supp}(e, \phi(\alpha^*)). \tag{16}$$

The fitness function that measures the quality of a partition (subgroup) S_d is defined as follows:

$$\text{fitness}(S_d) = \frac{\sum_{e \subset S_d} \text{Weight}(S_d)}{\sum_{|e \cap S_d| > 0} \text{Weight}(S_d)}. \tag{17}$$

The fitness function measures the ratio of weights of hyperedges that are within the partition and weights of hyperedges involving any vertex of this partition. High fitness value suggests that vertices within the partition are more connected to each other than to other vertices.

We find the largest number of subgroups partitioned using hMETIS while the fitness function of each subgroup stays above a given threshold value. In the example, the smallest number of subgroups is 2, and the best partition given by hMETIS is: $\{1, 2, 3\}$ and $\{4, 5, 6\}$. The fitness values of the two subgroups are 1, which is the highest possible value of the fitness function. If we increase the number of subgroups to 3, then the best partition is $\{1\}$, $\{2, 3\}$, and $\{4, 5, 6\}$. The fitness values of the three subgroups are 0, 0.25, and 1. So, it is clear that the best number of subgroups should be 2.

[1] A frequent individual item is an item that appears sufficiently often through time.

[2] A frequent item set is an item set whose items simultaneously appear sufficiently often through time.

TABLE II
$s(\phi(\alpha^*), k, t)$ FOR EIGHT AGENTS AND EIGHT TIME POINTS

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8
Agent 1	1	0	0	0	0	0	0	0
Agent 2	0	0	0	1	1	0	0	1
Agent 3	0	1	0	0	0	1	1	0
Agent 4	0	0	1	0	0	0	0	0
Agent 5	1	0	0	1	1	0	1	0
Agent 6	0	1	1	1	1	1	1	0
Agent 7	1	1	0	0	0	1	0	1
Agent 8	0	0	1	0	0	0	0	1

2) Group Partition Based on Complementarity: In a complementarity relationship, the number of agents in a subgroup that perform the task is expected to be as constant as possible. For example, the proposition "at least 40 and at most 50 agents from a subgroup of 100 agents should perform the task" is deemed more precise than "at least 10 and at most 90 agents from a subgroup of 100 agents should perform the task." One good measure of how far a set of numbers are spread out is the variance. If a subgroup e of agents acts complementarily, then the variance of the number of agents in subgroup e that perform the task at different time points should be small.

We still transform the clustering problem to a hypergraph-partitioning problem. The partitioning procedure and the definition of fitness functions are the same as the similarity relationship approach. The only differences are that we assign every possible set e of vertices as a hyperedge and the weight of hyperedge e is defined as follows:

$$\text{Weight}(e) = 1/(\text{Var}(e) + \epsilon) \tag{18}$$

where ϵ is a small positive number such as 10^{-7} to avoid singularity in the case of $\text{Var}(e) = 0$, and the variance $\text{Var}(e)$ is defined as

$$\text{Var}(e) = \sum_{q=1}^{z} \sum_{j=1}^{l_{\phi(\alpha^*),q}} \left(\sum_{k=1}^{n} \mathcal{X}_{k,e} s_q(\phi(\alpha^*), k, j) - \frac{1}{\sum_{q=1}^{z} l_{\phi(\alpha^*),q}} \times \sum_{q=1}^{z} \sum_{j=1}^{l_{\phi(\alpha^*),q}} \sum_{k=1}^{n} \mathcal{X}_{k,e} s_q(\phi(\alpha^*), k, j) \right)^2 \Bigg/ \sum_{q=1}^{z} l_{\phi(\alpha^*),q} \tag{19}$$

where $\mathcal{X}_{k,e}$ is a binary variable that describes whether vertex (agent) k belongs to hyperedge e, i.e., $\mathcal{X}_{k,e} = 1$ if vertex (agent) k belongs to hyperedge e and $\mathcal{X}_{k,e} = 0$ otherwise.

In the furniture moving scenario, we give another example of eight agents and one observation trajectory of eight time points (representing eight consecutive hours) for each agent with the signature $s(\phi(\alpha^*), k, t)$ listed in Table II. We start from the smallest number of subgroups and the best partition given by hMETIS is: $\{1, 2, 3, 4\}$ and $\{5, 6, 7, 8\}$. The fitness values of the two subgroups are 0.7354 and 0.7364. If we increase the number of subgroups to 3, then the best partition is $\{7\}$, $\{1, 2, 3, 4\}$, and $\{5, 6, 8\}$. The fitness values of the three subgroups are 0, 0.7354, and 0.5789. So, the best number of subgroups is 2. It can be seen from Table II that there are always one agent from $\{1, 2, 3, 4\}$ and two agents from $\{5, 6, 7, 8\}$ that are satisfying $\phi(\alpha^*)$ at any time point.

D. "Outer Logic" CensusSTL Formula Inference

After partitioning the group into several subgroups, we can proceed to generate the "*outer logic*" CensusSTL formula from the census trajectories.

We denote all parameters that define the "*outer logic*" formula γ as β. The inference of the "*outer logic*" CensusSTL formula is also a constrained multiobjective optimization problem for finding the best parameters β that describe the formula γ, and we use particle swarm optimization to find β. In the inference of the "outer logic" formula, we specify the "*outer logic*" formula γ to be in the form of $\gamma = \Box_{[0,T_\gamma)}(\gamma_c \Rightarrow \gamma_e)$, with γ_c being the cause and γ_e being the effect formula. In this form, we can capture causal relationships that are maintained consistently during a time period. All the temporal parameters of γ_c and γ_e chosen in $\mathbb{R}_{\geqslant 0}$ [T_γ is the length of the time domain of the formula $(\gamma_c \Rightarrow \gamma_e)$ with respect to the census trajectories].

We consider eight templates of temporal logic formula γ.

1) Instantaneous Cause Durational Effect:

$$\gamma = \Box_{[0,T_c-\tau_2)}(\gamma_{cs} \Rightarrow \Box_{[\tau_1,\tau_2)}\gamma_{es}) \tag{20}$$

where T_c is the length of the census trajectories, and γ_{cs} and γ_{es} are the cause and effect CensusSTL formulas without temporal operators. This causal relationship means "*during the next $T_c - \tau_2$ time units, whenever γ_{cs} is true, then γ_{es} will always be true from the next τ_1 to τ_2 time units.*"

2) Instantaneous Cause Eventual Effect:

$$\gamma = \Box_{[0,T_c-\tau_2)}(\gamma_{cs} \Rightarrow \Diamond_{[\tau_1,\tau_2)}\gamma_{es}) \tag{21}$$

which means "*during the next $T_c - \tau_2$ time units, whenever γ_{cs} is true, then γ_{es} will be true at least once from the next τ_1 to τ_2 time units.*"

3) Instantaneous Cause Eventual Durational Effect:

$$\gamma = \Box_{[0,T_c-\tau_2-\tau_3)}(\gamma_{cs} \Rightarrow \Diamond_{[\tau_1,\tau_2)}\Box_{[0,\tau_3)}\gamma_{es}) \tag{22}$$

which means "*during the next $T_c - \tau_2 - \tau_3$ time units, whenever γ_{cs} is true, then γ_{es} will be true at least once from the next τ_1 to τ_2 time units and maintain to be true for τ_3 time units.*"

4) Instantaneous Cause Persistent Effect:

$$\gamma = \Box_{[0,T_c-\tau_2-\tau_3)}(\gamma_{cs} \Rightarrow \Box_{[\tau_1,\tau_2)}\Diamond_{[0,\tau_3)}\gamma_{es}) \tag{23}$$

which means "*during the next $T_c - \tau_2 - \tau_3$ time units, whenever γ_{cs} is true, then from the next τ_1 to τ_2 time units, γ_{es} will be true at least once every τ_3 time units.*"

5) Durational Cause Durational Effect:

$$\gamma = \Box_{[0,T_c-\tau_3)}(\Box_{[0,\tau_1)}\gamma_{cs} \Rightarrow \Box_{[\tau_2,\tau_3)}\gamma_{es}) \tag{24}$$

which means "*during the next $T_c - \tau_3$ time units, whenever γ_{cs} is true for τ_1 time units, then γ_{es} will always be true from τ_2 to τ_3 time units.*"

6) Durational Cause Eventual Effect:

$$\gamma = \Box_{[0,T_c-\tau_3)}(\Box_{[0,\tau_1)}\gamma_{cs} \Rightarrow \Diamond_{[\tau_2,\tau_3)}\gamma_{es}) \tag{25}$$

which means "*during the next $T_c - \tau_3$ time units, whenever γ_{cs} is true for τ_1 time units, then γ_{es} will be true at least once from τ_2 to τ_3 time units.*"

7) Durational Cause Eventual Durational Effect:

$$\gamma = \Box_{[0,T_c-\tau_3-\tau_4)}(\Box_{[0,\tau_1)}\gamma_{cs} \Rightarrow \Diamond_{[\tau_2,\tau_3)}\Box_{[0,\tau_4)}\gamma_{es}) \quad (26)$$

which means "*during the next $T_c-\tau_3-\tau_4$ time units, whenever γ_{cs} is true for τ_1 time units, then γ_{es} will be true at least once from τ_2 to τ_3 time units and maintain to be true for τ_4 time units*."

8) Durational Cause Persistent Effect:

$$\gamma = \Box_{[0,T_c-\tau_3-\tau_4)}(\Box_{[0,\tau_1)}\gamma_{cs} \Rightarrow \Box_{[\tau_2,\tau_3)}\Diamond_{[0,\tau_4)}\gamma_{es}) \quad (27)$$

which means "*during the next $T_c-\tau_3-\tau_4$ time units, whenever γ_{cs} is true for τ_1 time units, then from τ_2 to τ_3 time units, γ_{es} will be true at least once every τ_4 time units*."

Definition 8: We define $m(\gamma)$ as the total number of time points at which the formula γ is true in the training data set (the time points in different census trajectories are counted separately).

Definition 9: We define $p(\gamma_c \Rightarrow \gamma_e)$ as the accuracy rate of the CensusSTL formula $\gamma = \Box_{[0,T_\gamma)}(\gamma_c \Rightarrow \gamma_e)$ in the training data set, and its value is calculated as follows:

$$p(\gamma_c \Rightarrow \gamma_e) = m(\gamma_c \wedge \gamma_e)/m(\gamma_c). \quad (28)$$

$p(\gamma_c \Rightarrow \gamma_e)$ is generally a number between 0 and 1, but in the case of $m(\gamma_c) = 0$, its value becomes infinity. To avoid this, we specify $p(\gamma_c \Rightarrow \gamma_e)$ to be -1 when $m(\gamma_c) = 0$ in the calculations of the objective functions introduced in the following.

The optimization has three objectives in general: the first objective is to maximize $100p(\gamma_c \Rightarrow \gamma_e)$, which is the percent value of the accuracy rate of the formula in the training data set; the second objective is to maximize $m(\gamma_c)$ so as to maximize the frequency of the formula γ_c in the training data set; the last objective is to make the formula γ more precise.

In particular, for similarity-based partitioning, we make the number of agents in a subgroup that perform the task as large as possible (ideally the same as the number of agents in the subgroup), so the optimization is formulated as follows:

$$\begin{aligned}
&\min \ -100p(\gamma_c(\beta) \Rightarrow \gamma_e(\beta)) - \lambda_1' m(\gamma_c(\beta)) - \lambda_2'(c_{i1} + c_{j2}) \\
&\text{s.t. } \gamma \in \Phi_\gamma \\
&\quad \gamma_{cs} = n(\phi, S_i) > c_{i1} (i = 1, 2, \ldots, n_s) \\
&\quad \gamma_{es} = n(\phi, S_j) > c_{j2} (j = 1, 2, \ldots, n_s)
\end{aligned}$$

where β (including the temporal parameters and c_{i1} and c_{j2}) is the optimization variable, n_s is the number of subgroups partitioned based on similarity, and λ_1' and λ_2' are the weighting factors (for tuning of λ_1' and λ_2', see the example in Section IV), Φ_γ is the set of the eight templates of γ. For any of the eight templates, there are n_s^2 different CensusSTL formula γ values as both γ_c (which contains γ_{cs}) and γ_e (which contains γ_{es}) can be about any of the n_s subgroups.

In the furniture moving scenario from Table I, one of the best formula obtained is as follows:

$$\Box_{[0,4)}(\Box_{[0,2)}(n(\phi, S_{s1}) > 2) \Rightarrow \Box_{[2,4)}(n(\phi, S_{s2}) > 2)) \quad (29)$$

which means "*for the next 4 h, whenever the three agents from $\{1, 2, 3\}$ move the furniture from Region 1 to Region 2 for 2 h, then the three agents from $\{4, 5, 6\}$ will be moving the furniture from Region 1 to Region 2 for the next 2 h.*"

For complementarity-based partitioning, we make the number of agents in a subgroup that perform the task as constant as possible; the optimization is formulated as follows:

$$\begin{aligned}
&\min \ -100p(\gamma_c(\beta) \Rightarrow \gamma_e(\beta)) - \lambda_1' m(\gamma_c(\beta)) \\
&\qquad + \lambda_2' \sum_{i=1}^{2n_c} (c_{i2}' - c_{i1}') \\
&\text{s.t. } \gamma \in \Phi_\gamma \\
&\quad \gamma_{cs} = \bigwedge_{i=1}^{n_c} (n(\phi, S_i) > c_{i1}' \wedge n(\phi, S_i) < c_{i2}') \\
&\quad \gamma_{es} = \bigwedge_{i=n_c+1}^{2n_c} (n(\phi, S_i) > c_{i1}' \wedge n(\phi, S_i) < c_{i2}') \\
&\quad c_{i2}' - c_{i1}' > 1 \quad (i = 1, 2, \ldots, 2n_c)
\end{aligned}$$

where β (including the temporal parameters and c_{i1}' and c_{i2}') is the optimization variable, n_c is the number of subgroups partitioned based on complementarity, and λ_1' and λ_2' are the weighting factors (for tuning of λ_1' and λ_2', see the example in Section IV).

In the furniture moving scenario from Table II, one of the best formula obtained is as follows:

$$\begin{aligned}
&\Box_{[0,6)}(n(\phi, S_{c1}) > 0 \wedge n(\phi, S_{c1}) < 2 \wedge n(\phi, S_{c2}) > 1 \\
&\quad \wedge n(\phi, S_{c2}) < 3 \Rightarrow \Box_{[0,2)}(n(\phi, S_{c1}) > 0 \wedge n(\phi, S_{c1}) \\
&\quad < 2 \wedge n(\phi, S_{c2}) > 1 \wedge n(\phi, S_{c2}) < 3))
\end{aligned} \quad (30)$$

which means "*for the next 6 h, whenever there are one agent from $\{1, 2, 3, 4\}$ and two agents from $\{5, 6, 7, 8\}$ who are moving the furniture from Region 1 to Region 2, then for the next 2 h, there will still be one agent from $\{1, 2, 3, 4\}$, and two agents from $\{5, 6, 7, 8\}$ who move the furniture from Region 1 to Region 2*."

E. CensusSTL Formula Validation

The obtained CensusSTL formula is validated in a separate validation data set. The accuracy rate of the formula $\gamma_c \Rightarrow \gamma_e$ in the validation data set is as follows:

$$p_v(\gamma_c \Rightarrow \gamma_e) = m_v(\gamma_c \wedge \gamma_e)/m_v(\gamma_c) \quad (31)$$

where $m_v(\gamma)$ is the total number of time points at which the CensusSTL formula γ is true in the validation data set.

IV. Implementation

In order to test the effectiveness of the algorithm, we consider a data set from a soccer match that happened on November 7th, 2013 between Tromsø IL (Norway) and Anzhi Makhachkala (Russia) at Alfheim Stadium, Tromsø, Norway. Tromsø IL will be referred to as the home team and the Anzhi Makhachkala as the visiting team. The players of the home team are equipped with body sensors during the whole game. The body-sensor data and video camera data of the players of the home team are provided in [32]. The x-axis points southward parallel with the long side of the field, while

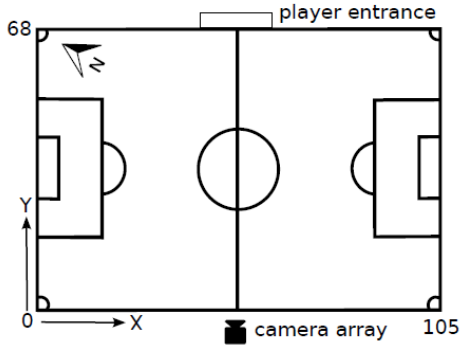

Fig. 2. Layout at the Alfheim Stadium [32].

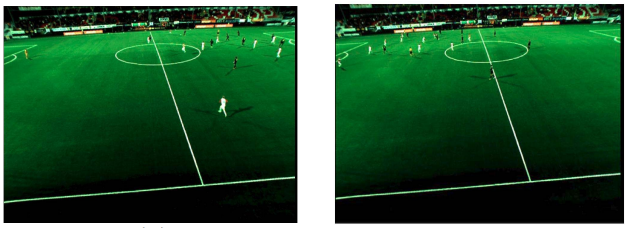

Fig. 3. Video capture at (a) 17 min 36 s and (b) 17 min 47 s.

the y-axis points eastward parallel with the short edge of the field, as shown in Fig. 2. The soccer pitch is 105×68 m, and hence, the values for x and y are in the range of $0 \leq x \leq 105$ and $0 \leq y \leq 68$ if the players are in the field.

We focus on the situation that a player in the home team is attacking in the visiting team's half field and then the player suddenly runs back to the home field. This usually happens, because the ball is intercepted by the visiting team who launches a counterattack and the players in the home team run back for defense. For example, at 17 min 36 s [as shown in Fig. 3(a)], many players are in the visiting team's half field, then at 17 min 47 s, most players run back to the home field [as shown in Fig. 3(b)]. We call it a runback situation and we want to derive a CensusSTL formula for the behaviors of different subgroups of the home team. As the runback task is a sequential task, we select the following STL formula to be the template for the "*inner logic*" formula:

$$\begin{aligned} \phi &= \Diamond_{[-\tau_3-\tau_4,0]}\phi_t \\ \phi_t &= \Box_{[0,\tau_1)}(p_{(\text{red region})}) \wedge \Diamond_{[\tau_2,\tau_3)}\Box_{[0,\tau_4)}(p_{(\text{yellow region})}) \end{aligned} \tag{32}$$

which reads "*the player stays in the red region for τ_1 seconds, then sometime between τ_2 and τ_3 seconds, he arrives in the yellow region and stay there for at least τ_4 seconds.*"

The *a priori* regions for the red region $[\mathcal{O}(X_1)]$ and yellow region $[\mathcal{O}(X_2)]$ are selected symmetrically in the home field and the visiting team's half field near the half-way line, as shown in Fig. 4(a). As the runback task generally takes less than 12 s, we set $\|\phi_t(\alpha)\| = \tau_3 + \tau_4 \leq \tau_{\text{limit}} = 12$ in the optimization process.

Considering that there were substitutions of players in the second half of the match, we focus on the first half of the game. The data for the positions of the ten outfield players (the goalkeeper is excluded) are discontinuous (some data are not available at certain time intervals). We choose to use the data from the longest time interval with continuous data from 16 min and 19 s to 29 min and 27 s (789 s in total, sampled at every second) as the training data set for CensusSTL formula inference and use the data from 5 min and 13 s to 9 min and 57 s (285 s in total) as the validation data set.

TABLE III
RESULTS WITH DIFFERENT λ_2 VALUES

	$\sum_{k=1}^{n} m(\phi(\alpha), k)$	$d_{\text{H}}(\mathcal{O}(X_1), \mathcal{O}(p_1(\alpha)))$	$d_{\text{H}}(\mathcal{O}(X_2), \mathcal{O}(p_2(\alpha)))$
$\lambda_2 = 1$	6788	986.7829	400
$\lambda_2 = 40$	968	5.5881	51.4695
$\lambda_2 = 100$	810	0.3743	0.5104

1) "Inner Logic" Formula Inference: In cost function (12), we set λ_1 to be 1, λ_2 to be 1, 40, and 100 and the results are listed in Table III. When $\lambda_2 = 1$, there are many times (6788 times) that the "*inner logic*" formula is true, but the obtained predicates are very different from the *a priori* predicates (Hausdorff distances being 986.7829 and 400). Besides, the red region and the yellow region overlap in the middle [as shown in Fig. 4(b)], which leads to very ambiguous result as the player can just stay in the overlapped region and not actually "run back." When $\lambda_2 = 100$, the obtained predicates are almost the same as the *a priori* predicates [as shown in Fig. 4(d), Hausdorff distances being 0.3743 and 0.5104], but there are much fewer times (810 times) the "*inner logic*" formula is true. In comparison, when $\lambda_2 = 40$, there are 968 times the "*inner logic*" formula is true and the obtained regions remain similar with the *a priori* regions with no overlap in the middle [as shown in Fig. 4(c)].

The obtained "*inner logic*" formula when $\lambda_1 = 1$ and $\lambda_2 = 40$ is as follows:

$$\begin{aligned} \phi &= \Diamond_{[-12,0]}(\Box_{[0,2)}(p_{(\text{red region})}(\alpha^*)) \\ &\qquad \wedge \Diamond_{[2,10)}\Box_{[0,2)}(p_{(\text{yellow region})}(\alpha^*))) \\ p_{(\text{red region})}(\alpha^*) &= (0.99873x + 0.050441y > 57.2938) \\ &\quad \wedge (0.36068x - 0.93269y < 67.2938) \\ &\quad \wedge (0.91245x + 0.40919y > 19.7634) \wedge (x < 86.3008) \\ p_{(\text{yellow region})}(\alpha^*) &= (0.97221x + 0.23409y > 21.7704) \\ &\quad \wedge (0.87666x - 0.48111y < 81.3772) \\ &\quad \wedge (0.93448x - 0.35601y > -2.159) \\ &\quad \wedge (0.99436x - 0.10605y < 47.2862). \end{aligned} \tag{33}$$

2) Group Partition: In the second step, we partition the group based on the satisfaction signature trajectories of the ten players with respect to ϕ. The number of players whose behaviors satisfy ϕ in the team is shown in Fig. 5(a). The satisfaction signatures of the ten players with respect to ϕ are shown in Fig. 5(b). We first partition the group

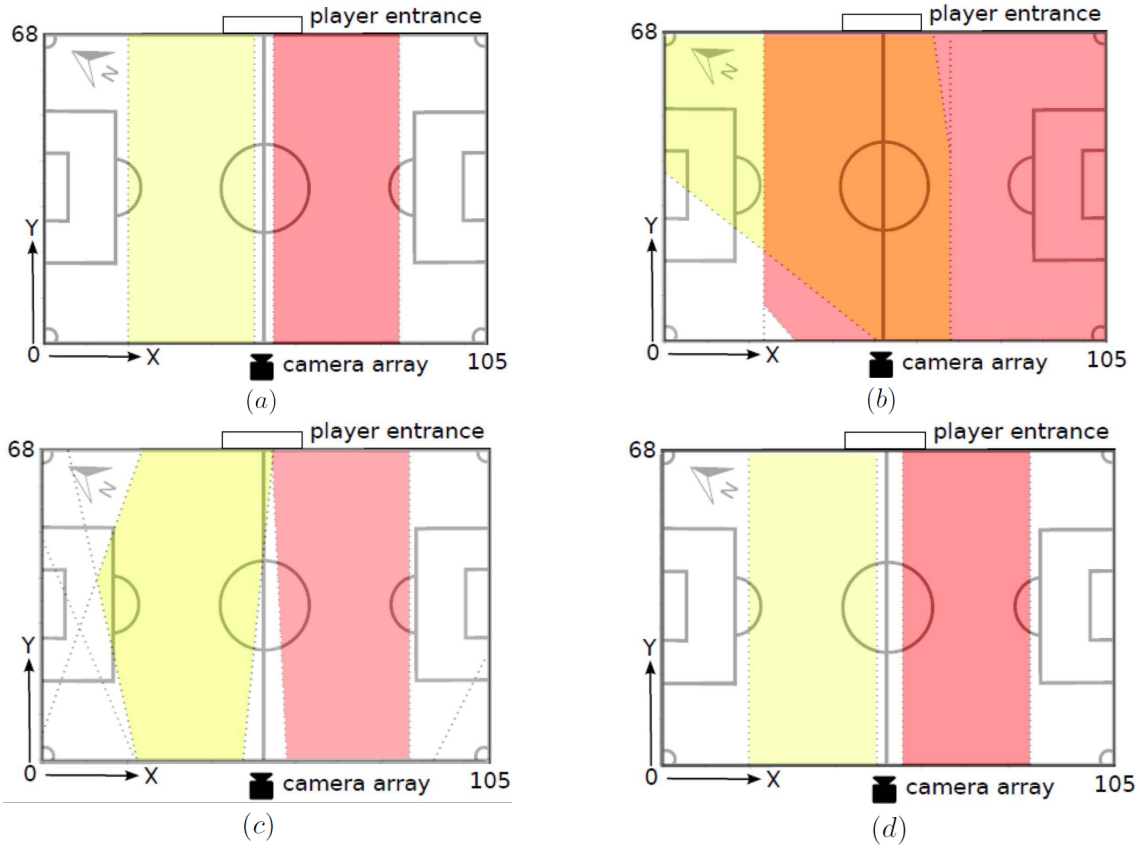

Fig. 4. Alfheim Stadium with (a) *a priori* regions, (b) obtained regions with $\lambda_2 = 1$, (c) obtained regions with $\lambda_2 = 40$, and (d) obtained regions with $\lambda_2 = 100$.

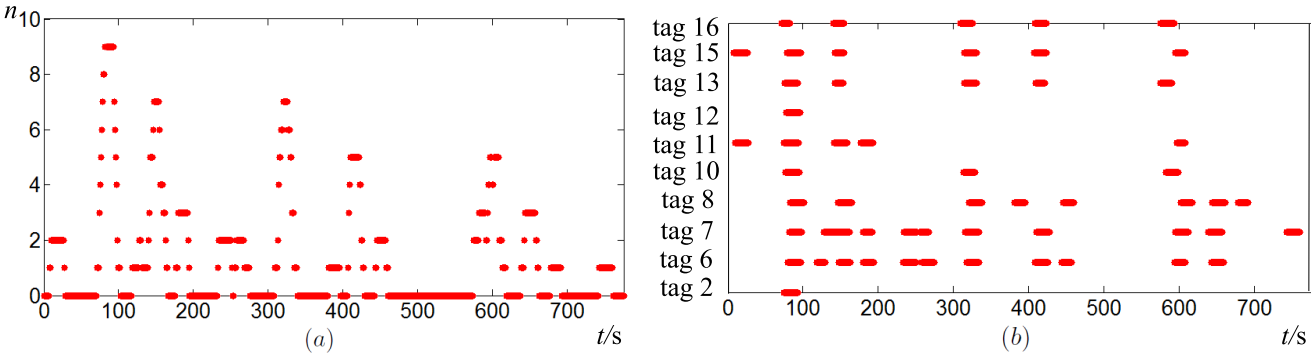

Fig. 5. (a) Number of players whose behaviors satisfy ϕ in the training set. (b) Satisfaction signatures of the ten players in the training set with respect to ϕ (red dots represent that the signature equals one).

based on similarity. It can be seen from Fig. 5(b) that the players of tag2, tag10, and tag12 do not perform the runback task as frequently as the other players, so we set *minsup* to be 0.1 to exclude the three players. With $S_f = \{\text{tag6, tag7, tag8, tag11, tag13, tag15, tag16}\}$, the partition results are shown in Table IV. We set the threshold for the fitness function of a good partition to be 0.2, and the largest number of subgroups that can satisfy the criterion is 3.

We then partition the group based on complementarity and the partition results are shown in Table V. With the same threshold for the fitness function as 0.2, the largest number of subgroups that can satisfy the criterion is 2.

3) "Outer Logic" CensusSTL Formula Inference: We first identify the CensusSTL formulas for the subgroups partitioned based on similarity. We find the best formula for each of the eight different templates. As $n_s = 3$, there are $3^2 = 9$ different formulas for any of the eight different templates. In this paper, due to space limitations, we only infer the formula with $\gamma_{cs} = n(\phi, S_1) > c_{11}$ and $\gamma_{es} = n(\phi, S_2) > c_{22}$, and the other eight types of formulas can be inferred similarly. We set λ_1' in the objective function to be 1, for the value of λ_2', we take γ_2 as an example and obtained different results as listed in Table VI. When $\lambda_2' = 0.1$, we obtain a formula with 80.81% accuracy rate, but the formula is not very precise

TABLE IV
PARTITION RESULTS BASED ON SIMILARITY

	$fitness(S_{s1})$	$fitness(S_{s2})$	$fitness(S_{s3})$	$fitness(S_{s4})$
$S_{s1} = \{tag11, tag13, tag16\}$ $S_{s2} = \{tag6, tag7, tag8, tag15\}$	0.3166	0.8270	NA	NA
$S_{s1} = \{tag13, tag16\}$ $S_{s2} = \{tag6, tag7, tag8\}$ $S_{s3} = \{tag11, tag15\}$	0.4387	0.6943	0.2421	NA
$S_{s1} = \{tag6, tag7, tag8\}$ $S_{s2} = \{tag13, tag16\}$ $S_{s3} = \{tag11\}$ $S_{s4} = \{tag15\}$	0.6943	0.4387	0	0

TABLE V
PARTITION RESULTS BASED ON COMPLEMENTARITY

	$fitness(S_{c1})$	$fitness(S_{c2})$	$fitness(S_{c3})$	$fitness(S_{c4})$
$S_{c1} = \{tag6, tag7, tag13, tag15\}$ $S_{c2} = \{tag2, tag8, tag10, tag11, tag12, tag16\}$	0.4256	0.6776	NA	NA
$S_{c1} = \{tag6, tag8, tag11, tag13\}$ $S_{c2} = \{tag2, tag10, tag12, tag16\}$ $S_{c3} = \{tag7, tag15\}$	0.4205	0.5241	0.1743	NA

TABLE VI
RESULTS WITH DIFFERENT λ_2' VALUES

	γ_1	$m(\gamma_c \wedge \gamma_e)$	$m(\gamma_c)$	$p(\gamma_c \Rightarrow \gamma_e)$
$\lambda_2' = 0.1$	$\Box_{[0,780)}(n(\phi, S_{s1}) > 0 \Rightarrow \Box_{[7,9)} n(\phi, S_{s2}) > 0)$	80	99	80.81%
$\lambda_2' = 1$	$\Box_{[0,779)}(n(\phi, S_{s1}) > 0 \Rightarrow \Box_{[8,10)} n(\phi, S_{s2}) > 1)$	65	99	67.68%
$\lambda_2' = 10$	$\Box_{[0,779)}(n(\phi, S_{s1}) > 1 \Rightarrow \Box_{[8,10)} n(\phi, S_{s2}) > 2)$	34	65	52.31%

TABLE VII
RESULTS BASED ON SIMILARITY

	$m(\gamma_c \wedge \gamma_e)$	$m(\gamma_c)$	$p(\gamma_c \Rightarrow \gamma_e)$	$m_v(\gamma_c \wedge \gamma_e)$	$m_v(\gamma_c)$	$p_v(\gamma_c \Rightarrow \gamma_e)$
$\gamma_1 = \Box_{[0,780)}(n(\phi, S_{s1}) > 0 \Rightarrow \Box_{[7,9)} n(\phi, S_{s2}) > 0)$	80	99	80.81%	21	42	50%
$\gamma_2 = \Box_{[0,755)}(n(\phi, S_{s1}) > 0 \Rightarrow \Diamond_{[1,34)} n(\phi, S_{s2}) > 1)$	99	99	100%	42	42	100%
$\gamma_3 = \Box_{[0,738)}(n(\phi, S_{s1}) > 0 \Rightarrow \Diamond_{[1,50)} \Box_{[0,1)} n(\phi, S_{s2}) > 1)$	99	99	100%	40	42	95.24%
$\gamma_4 = \Box_{[0,776)}(n(\phi, S_{s1}) > 0 \Rightarrow \Box_{[2,3)} \Diamond_{[0,10)} n(\phi, S_{s2}) > 1)$	91	99	91.92%	22	42	52.38%
$\gamma_5 = \Box_{[0,717)}(\Box_{[0,2)} n(\phi, S_{s1}) > 1 \Rightarrow \Box_{[70,72)} n(\phi, S_{s2}) > 1)$	74	89	83.15%	19	38	50%
$\gamma_6 = \Box_{[0,739)}(\Box_{[0,2)} n(\phi, S_{s1}) > 0 \Rightarrow \Diamond_{[1,50)} n(\phi, S_{s2}) > 1)$	94	94	100%	40	40	100%
$\gamma_7 = \Box_{[0,766)}(\Box_{[0,2)} n(\phi, S_{s1}) > 1 \Rightarrow \Diamond_{[1,20)} \Box_{[0,3)} n(\phi, S_{s2}) > 1)$	89	89	100%	27	38	71.05%
$\gamma_8 = \Box_{[0,777)}(\Box_{[0,3)} n(\phi, S_{s1}) > 0 \Rightarrow \Box_{[5,7)} \Diamond_{[0,5)} n(\phi, S_{s2}) > 1)$	75	89	100%	18	38	47.37%

as c_{11} and c_{22} are relatively small ($c_{11} = 0$ and $c_{22} = 0$) compared with the number of players in the subgroups. When $\lambda_2' = 1$, we obtain a formula, which is more precise ($c_{11} = 0$ and $c_{22} = 1$), but the accuracy rate of the formula drops to 67.68%. When $\lambda_2' = 10$, we obtain a formula, which is the most precise ($c_{11} = 1$ and $c_{22} = 2$), and the accuracy

TABLE VIII
RESULTS BASED ON COMPLEMENTARITY

	m $(\gamma_c \wedge \gamma_e)$	$m(\gamma_c)$	p $(\gamma_c \Rightarrow \gamma_e)$	m_v $(\gamma_c \wedge \gamma_e)$	$m_v(\gamma_c)$	p_v $(\gamma_c \Rightarrow \gamma_e)$
$\gamma_1' = \square_{[0,719)}(n(\phi, S_{c1}) > 0 \wedge n(\phi, S_{c1}) < 4 \wedge n(\phi, S_{c2}) > 0 \wedge n(\phi, S_{c2}) < 6 \Rightarrow \square_{[68,70)}(n(\phi, S_{c1}) > 0 \wedge n(\phi, S_{c1}) < 4 \wedge n(\phi, S_{c2}) > 0 \wedge n(\phi, S_{c2}) < 6))$	110	138	79.71%	34	42	80.95%
$\gamma_2' = \square_{[0,767)}(n(\phi, S_{c1}) > 0 \wedge n(\phi, S_{c1}) < 4 \wedge n(\phi, S_{c2}) > 0 \wedge n(\phi, S_{c2}) < 3 \Rightarrow \Diamond_{[1,22)}(n(\phi, S_{c1}) > 0 \wedge n(\phi, S_{c1}) < 2 \wedge n(\phi, S_{c2}) > 0 \wedge n(\phi, S_{c2}) < 2))$	128	128	100%	36	36	100%
$\gamma_3' = \square_{[0,687)}(n(\phi, S_{c1}) > 0 \wedge n(\phi, S_{c1}) < 4 \wedge n(\phi, S_{c2}) > 0 \wedge n(\phi, S_{c2}) < 4 \Rightarrow \Diamond_{[0,99)} \square_{[0,3)}(n(\phi, S_{c1}) > 0 \wedge n(\phi, S_{c1}) < 2 \wedge n(\phi, S_{c2}) > 0 \wedge n(\phi, S_{c2}) < 2))$	126	132	95.45%	36	36	100%
$\gamma_4' = \square_{[0,687)}(n(\phi, S_{c1}) > 0 \wedge n(\phi, S_{c1}) < 4 \wedge n(\phi, S_{c2}) > 0 \wedge n(\phi, S_{c2}) < 4 \Rightarrow \square_{[0,2)} \Diamond_{[0,100)}(n(\phi, S_{c1}) > 0 \wedge n(\phi, S_{c1}) < 2 \wedge n(\phi, S_{c2}) > 0 \wedge n(\phi, S_{c2}) < 2))$	128	132	96.97%	32	32	100%
$\gamma_5' = \square_{[0,719)}(\square_{[0,1)} n(\phi, S_{c1}) > 0 \wedge n(\phi, S_{c1}) < 4 \wedge n(\phi, S_{c2}) > 0 \wedge n(\phi, S_{c2}) < 6 \Rightarrow \square_{[68,70)}(n(\phi, S_{c1}) > 0 \wedge n(\phi, S_{c1}) < 4 \wedge n(\phi, S_{c2}) > 0 \wedge n(\phi, S_{c2}) < 6))$	124	124	100%	38	38	100%
$\gamma_6' = \square_{[0,691)}(\square_{[0,1)} n(\phi, S_{c1}) > 0 \wedge n(\phi, S_{c1}) < 4 \wedge n(\phi, S_{c2}) > 0 \wedge n(\phi, S_{c2}) < 6 \Rightarrow \Diamond_{[5,98)}(n(\phi, S_{c1}) > 0 \wedge n(\phi, S_{c1}) < 2 \wedge n(\phi, S_{c2}) > 0 \wedge n(\phi, S_{c2}) < 2))$	118	124	95.16%	34	34	100%
$\gamma_7' = \square_{[0,664)}(\square_{[0,2)} n(\phi, S_{c1}) > 0 \wedge n(\phi, S_{c1}) < 4 \wedge n(\phi, S_{c2}) > 0 \wedge n(\phi, S_{c2}) < 6 \Rightarrow \Diamond_{[5,124)} \square_{[0,1)}(n(\phi, S_{c1}) > 0 \wedge n(\phi, S_{c1}) < 2 \wedge n(\phi, S_{c2}) > 0 \wedge n(\phi, S_{c2}) < 2))$	117	117	100%	30	30	100%
$\gamma_8' = \square_{[0,667)}(\square_{[0,2)} n(\phi, S_{c1}) > 0 \wedge n(\phi, S_{c1}) < 4 \wedge n(\phi, S_{c2}) > 0 \wedge n(\phi, S_{c2}) < 6 \Rightarrow \square_{[5,22)} \Diamond_{[0,100)}(n(\phi, S_{c1}) > 0 \wedge n(\phi, S_{c1}) < 2 \wedge n(\phi, S_{c2}) > 0 \wedge n(\phi, S_{c2}) < 2))$	114	124	97.14%	34	34	100%

rate of the formula drops to 52.31%. While, in general, the user can choose a formula with higher accuracy rate or higher precision based on the preferences of the user, we choose the formulas with higher accuracy rates in such tradeoffs in this paper.

The CensusSTL formulas for the subgroups partitioned based on similarity are listed in Table VII. γ_1, γ_4, and γ_5 do not have 100% accuracy rate in the training data set and get even lower accuracy rates in the validation data set, so they are not the best formulas for this case. γ_7 and γ_8 all have 100% accuracy rate in the training data set, but they drop to lower accuracy rates in the validation data set. In comparison, γ_2, γ_3, and γ_6 have good performance in both the training data set and the validation data set, so they are the best formulas for similarity relationships.

γ_2 reads "*whenever there are at least one player from* {*tag*13, *tag*16} *who is running back, then sometime between the next 1 s and the next 34 s, at least two players from* {*tag*6, *tag*7, *tag*8} *will be running back*."

γ_3 reads "*whenever there are at least one player from* {*tag*13, *tag*16} *who is running back, then sometime between the next 1 s and the next 50 s, at least two players from* {*tag*6, *tag*7, *tag*8} *will be running back for at least 1 s*."

γ_6 reads "*whenever there are at least one player from* {*tag*13, *tag*16} *is running back for 2 s, then sometime between the next 1 s and the next 50 s, at least two players from* {*tag*6, *tag*7, *tag*8} *will be running back*."

Next, we identify the CensusSTL formulas for the subgroups partitioned based on complementarity. The obtained CensusSTL formulas for the eight different templates are listed in Table VIII. γ_1' is the only formula that does not have a 100% accuracy rate in the validation data set and is not accurate enough in the training data set as well. The rest of the formulas have 100% accuracy rates in the validation data set and have over 95% accuracy rates in the training data set, so they are all good formulas for this case. Among them, γ_2', γ_5' and γ_7' have 100% accuracy rates in both the training data set and the validation data set, so they are the best formulas for complementarity relationships. Although γ_5' is not very precise, it still provides useful information in a relatively strong formula structure (bounded always for both cause and effect formula).

γ_2' reads "*whenever there are at least one and at most three players from* {*tag*6, *tag*7, *tag*13, *tag*15} *and at least one and at most two players from* {*tag*2, *tag*8, *tag*10, *tag*11, *tag*12, *tag*16} *who are running back, then sometime between the next*

1 s and the next 22 s, there will still be exactly one player from $\{tag6, tag7, tag13, tag15\}$ *and one player from* $\{tag2, tag8, tag10, tag11, tag12, tag16\}$ *who are running back*."

γ_5' reads "*whenever there are at least one and at most three players from* $\{tag6, tag7, tag13, tag15\}$ *and at least one and at most five players from* $\{tag2, tag8, tag10, tag11, tag12, tag16\}$ *who are running back for at least 1 s, then from the next 68 s to the next 70 s, there will still be at least one and at most three players from* $\{tag6, tag7, tag13, tag15\}$ *and at least one and at most five players from* $\{tag2, tag8, tag10, tag11, tag12, tag16\}$ *who are running back*."

γ_7' reads "*whenever there are at least one and at most three players from* $\{tag6, tag7, tag13, tag15\}$ *and at least one and at most five players from* $\{tag2, tag8, tag10, tag11, tag12, tag16\}$ *who are running back for at least 2 s, then sometime between the next 5 s and the next 124 s, there will still be exactly one player from* $\{tag6, tag7, tag13, tag15\}$ *and one player from* $\{tag2, tag8, tag10, tag11, tag12, tag16\}$ *who are running back for at least 1 s*."

In Section V, we obtain some useful CensusSTL formulas in both similarity and complementarity relationship forms. The choice of the best formula depends not only on the performance in the training and validation data set, but also on the user preferences.

On a Dell desktop computer with a 3.20-GHz Intel Xeon CPU and 8-GB RAM, the "*inner logic*" formula inference took 881.2 s, the group partition took 2 s, and the "*outer logic*" formula inference took 72.4 s.

V. Conclusion

In this paper, we develop a novel formal framework for analyzing group behaviors using the newly defined CensusSTL. We used an inference algorithm to identify subgroups and find the CensusSTL formulas for different subgroups of a multiagent system. The inference algorithm is composed of three parts: 1) "*inner logic*" formula inference; 2) group partition based on complementarity and similarity; and 3) "*outer logic*" CensusSTL formula inference. Using the trajectories generated from the training set, the algorithm can discover new temporal–spatial properties about the structure of the system. We apply the algorithm in analyzing a soccer match, but similar approach can be used in the recommender systems, biological systems, MARS, monitoring systems, etc.

Acknowledgment

The authors would like to thank D. Johansen and his colleagues with the University of Tromsø for providing the data from the soccer match that we use in this paper, and C. Belta for introducing them to temporal logic inference.

References

[1] B. Guo, Z. Li, and S. Xu, "On modeling a soccer robot system using Petri nets," in *Proc. IEEE Int. Conf. Autom. Sci. Eng.*, Oct. 2006, pp. 460–465.

[2] X. Qiao *et al.*, "Recommending nearby strangers instantly based on similar check-in behaviors," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 3, pp. 1114–1124, Jul. 2015.

[3] H. Gao, J. Tang, X. Hu, and H. Liu, "Exploring temporal effects for location recommendation on location-based social networks," in *Proc. ACM Conf. Recommender Syst.*, New York, NY, USA, 2013, pp. 93–100.

[4] A. Zimdars, D. M. Chickering, and C. Meek, "Using temporal data for making recommendations," in *Proc. Conf. Uncertainty Artif. Intell.*, San Francisco, CA, USA, 2001, pp. 580–588.

[5] T. Shibata and T. Fukuda, "Coordinative behavior in evolutionary multi-agent-robot system," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 1, Jul. 1993, pp. 448–453.

[6] A. V. Timofeev, F. A. Kolushev, and A. A. Bogdanov, "Hybrid algorithms of multi-agent control of mobile robots," in *Proc. Int. Joint Conf. Neural Netw.*, vol. 6, 1999, pp. 4115–4118.

[7] H. Vermaak and J. Kinyua, "Multi-agent systems based intelligent maintenance management for a component-handling platform," in *Proc. IEEE Int. Conf. Autom. Sci. Eng.*, Sep. 2007, pp. 1057–1062.

[8] A. Fagiolini, G. Valenti, L. Pallottino, G. Dini, and A. Bicchi, "Local monitor implementation for decentralized intrusion detection in secure multi-agent systems," in *Proc. IEEE Int. Conf. Autom. Sci. Eng.*, Sep. 2007, pp. 454–459.

[9] D. E. Hernández-Mendoza, G. R. Peñaloza-Mendoza, and E. Aranda-Bricaire, "Discrete-time formation and marching control of multi-agent robots systems," in *Proc. Int. Conf. Elect. Eng. Comput. Sci. Autom. Control (CCE)*, 2011, pp. 1–6.

[10] C. Wang, L. Cao, and C. H. Chi, "Formalization and verification of group behavior interactions," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 8, pp. 1109–1124, Aug. 2015.

[11] K. T. Seow, "Integrating temporal logic as a state-based specification language for discrete-event control design in finite automata," *IEEE Trans. Autom. Sci. Eng.*, vol. 4, no. 3, pp. 451–464, Jul. 2007.

[12] Q. Jiang, "An improved algorithm for coordination control of multi-agent system based on r-limited Voronoi partitions," in *Proc. IEEE Int. Conf. Autom. Sci. Eng.*, Oct. 2006, pp. 667–671.

[13] J. Fu, H. G. Tanner, and J. Heinz, "Concurrent multi-agent systems with temporal logic objectives: Game theoretic analysis and planning through negotiation," *IET Control Theory Appl.*, vol. 9, no. 3, pp. 465–474, 2015.

[14] I. Filippidis, D. V. Dimarogonas, and K. J. Kyriakopoulos, "Decentralized multi-agent control from local LTL specifications," in *Proc. IEEE Conf. Decision Control*, Dec. 2012, pp. 6235–6240.

[15] S. Konur, M. Fisher, and S. Schewe, "Combined model checking for temporal, probabilistic, and real-time logics," *Theor. Comput. Sci.*, vol. 503, pp. 61–88, Sep. 2013.

[16] R. Alur, T. A. Henzinger, and O. Kupferman, "Alternating-time temporal logic," *J. ACM*, vol. 49, no. 5, pp. 672–713, 2002.

[17] M. Guo, J. Tumova, and D. V. Dimarogonas, "Cooperative decentralized multi-agent control under local LTL tasks and connectivity constraints," in *Proc. IEEE Conf. Decision Control*, Dec. 2014, pp. 75–80.

[18] T. Wongpiromsarn, A. Ulusoy, C. Belta, E. Frazzoli, and D. Rus, "Incremental synthesis of control policies for heterogeneous multi-agent systems with linear temporal logic specifications," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 5011–5018.

[19] Z. Xu, C. Belta, and A. Julius, "Temporal logic inference with prior information: An application to robot arm movements," in *Proc. IFAC Conf. Anal. Design Hybrid Syst. (ADHS)*, 2015, pp. 141–146.

[20] E. Asarin, A. Donzé, O. Maler, and D. Nickovic, "Parametric identification of temporal properties," in *Proc. 2nd Int. Conf. Runtime Verification*, Berlin, Germany, 2012, pp. 147–160.

[21] Z. Kong, A. Jones, A. M. Ayala, E. A. Gol, and C. Belta, "Temporal logic inference for classification and prediction from data," in *Proc. 17th Int. Conf. Hybrid Syst., Comput. Control*, New York, NY, USA, 2014, pp. 273–282.

[22] A. Jones, Z. Kong, and C. Belta, "Anomaly detection in cyber-physical systems: A formal methods approach," in *Proc. IEEE Conf. Decision Control*, Dec. 2014, pp. 848–853.

[23] J. R. Weeks, *Population: An Introduction to Concepts and Issues*. Belmont, CA, USA: Wadsworth, 1992. [Online]. Available: https://books.google.com/books?id=oIKMQgAACAAJ

[24] A. Donzé and O. Maler, "Robust satisfaction of temporal logic over real-valued signals," in *Proc. 8th Int. Conf. Formal Modeling Anal. Timed Syst.*, Berlin, Germany, 2010, pp. 92–106.

[25] M. J. Atallah, "A linear time algorithm for the Hausdorff distance between convex polygons," *Inf. Process. Lett.*, vol. 17, no. 4, pp. 207–209, 1983.

[26] F. Zhang, J. Cao, and Z. Xu, "An improved particle swarm optimization particle filtering algorithm," in *Proc. Int. Conf. Commun., Circuits Syst.*, vol. 2, 2013, pp. 173–177.

[27] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics: A Foundation for Computer Science*. Boston, MA, USA: Addison-Wesley, 1994.
[28] W. Sun, J. Wang, and Y. Fang, "Regularized k-means clustering of high-dimensional data and its asymptotic consistency," *Electron. J. Statist.*, vol. 6, pp. 148–167, 2012.
[29] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 359–392, Dec. 1998. [Online]. Available: http://dx.doi.org/10.1137/S1064827595287997
[30] E.-H. Han, G. Karypis, V. Kumar, and B. Mobasher, "Clustering in a high-dimensional space using hypergraph models," in *Proc. Data Mining Knowl. Discovery*, 1997. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.46.5972&rep=rep1&type=pdf
[31] T. Xu and X. Dong, "Mining frequent patterns with multiple minimum supports using basic *a priori*," in *Proc. Int. Conf. Natural Comput.*, 2013, pp. 957–961.
[32] S. A. Pettersen *et al.*, "Soccer video and player position dataset," in *Proc. 5th ACM Multimedia Syst. Conf.*, New York, NY, USA, 2014, pp. 18–23.

Zhe Xu (S'16) received the B.S. and M.S. degrees in electrical engineering from Tianjin University, Tianjin, China, in 2011 and 2014, respectively. He is currently pursuing the Ph.D. degree in electrical engineering with the Rensselaer Polytechnic Institute, Troy, NY, USA.

His current research interests include temporal logic, systems and control, hybrid systems, power systems, and mathematical control theory.

A. Agung Julius (M'06) received the Ph.D. degree in applied mathematics from the University of Twente, Enschede, The Netherlands, in 2005.

From 2005 to 2008, he was a Post-Doctoral Researcher with the University of Pennsylvania, Philadelphia, PA, USA. Since 2008, he has been with the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY, USA. He is currently an Associate Professor. His current research interests include systems and control, systems biology, stochastic models in systems biology, control of biological systems, hybrid systems, and mathematical systems theory.

Dr. Julius was a recipient of an NSF CAREER Award in 2010. He is an Associate Editor of the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING and the IFAC *Journal of Nonlinear Analysis: Hybrid Systems*.