

Probabilistic Event Calculus for Event Recognition

ANASTASIOS SKARLATIDIS^{1,2}, GEORGIOS PALIOURAS¹, ALEXANDER ARTIKIS^{3,1}
and GEORGE A. VOUIROS², ¹Institute of Informatics and Telecommunications, NCSR “Demokritos”,

²Department of Digital Systems, University of Piraeus

³Department of Maritime Studies, University of Piraeus

Symbolic event recognition systems have been successfully applied to a variety of application domains, extracting useful information in the form of events, allowing experts or other systems to monitor and respond when significant events are recognised. In a typical event recognition application, however, these systems often have to deal with a significant amount of uncertainty. In this paper, we address the issue of uncertainty in logic-based event recognition by extending the Event Calculus with probabilistic reasoning. Markov Logic Networks are a natural candidate for our logic-based formalism. However, the temporal semantics of the Event Calculus introduce a number of challenges for the proposed model. We show how and under what assumptions we can overcome these problems. Additionally, we study how probabilistic modelling changes the behaviour of the formalism, affecting its key property, the *inertia* of fluents. Furthermore, we demonstrate the advantages of the probabilistic Event Calculus through examples and experiments in the domain of activity recognition, using a publicly available dataset for video surveillance.

Categories and Subject Descriptors: I.2.3 [Deduction and Theorem Proving]: Uncertainty, “fuzzy,” and probabilistic reasoning; I.2.4 [Knowledge Representation Formalisms and Methods]: Temporal logic; I.2.6 [Learning]: Parameter learning; I.2.10 [Vision and Scene Understanding]: Video analysis

General Terms: Complex Event Processing, Event Calculus, Markov Logic Networks

Additional Key Words and Phrases: Events, Probabilistic Inference, Machine Learning, Uncertainty

ACM Reference Format:

Anastasios Skarlatidis, Georgios Paliouras, Alexander Artikis and George A. Vouros. 2014. Probabilistic Event Calculus for Event Recognition. *ACM Trans. Comput. Logic* V, N, Article A (January YYYY), 38 pages. DOI: <http://dx.doi.org/10.1145/0000000.0000000>

1. INTRODUCTION

Symbolic event recognition systems have received attention in a variety of application domains, such as health care monitoring, public transport management, telecommunication network monitoring and activity recognition [Luckham 2002; Etzion and Niblett 2010; Luckham 2011; Artikis et al. 2012; Turaga et al. 2008]. The aim of these systems is to extract useful information, in the form of events, by processing time-evolving data that comes from various sources (e.g. various types of sensor, surveillance cameras, network activity logs, etc.). The extracted information can be exploited by other systems or human experts, in order to monitor an environment and respond to the occur-

This work has been partially funded by the European Commission FP7 projects PRONTO (231738) and SPEEDD (619435). This paper is a significantly updated and extended version of Skarlatidis et al. [2011].

Author’s addresses: A. Skarlatidis and G. Paliouras, Institute of Informatics and Telecommunications, NCSR “Demokritos”, Athens 15310, Greece, {anskarl, paliourg}@iit.demokritos.gr; A. Artikis, Department of Maritime Studies, University of Piraeus, Piraeus 18534, Greece, a.artikis@iit.demokritos.gr; G. Vouros, Department of Digital Systems, University of Piraeus, Piraeus 18534, Greece, georgev@unipi.gr.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.

© YYYY ACM 1529-3785/YYYY/01-ARTA \$15.00

DOI: <http://dx.doi.org/10.1145/0000000.0000000>

rence of significant events. The input to a symbolic event recognition system consists of a stream of time-stamped symbols, called simple, derived events (SDEs). Consider, for example, a video tracking system detecting that someone is walking for a sequence of video frames. Based on such time-stamped input SDE observations, the symbolic event recognition system recognises composite events (CEs) of interest. For instance, that some people have started to move together. The recognition of a CE may be associated with the occurrence of various SDEs and other CEs involving multiple entities, e.g. people, vehicles, etc. CEs therefore, are relational structures over other sub-events, either CEs or SDEs.

Statistical approaches, e.g. probabilistic graphical models, employ Machine Learning techniques, in order to learn the situations under which CEs must be recognised from annotated examples. Such methods are data-driven and they are therefore dependent upon the examples of the training set. On the other hand, background knowledge (e.g. knowledge expressed by domain experts) may describe situations that do not appear in the training data or are difficult to be collected and annotated. The majority of statistical approaches employ models with limited capabilities in expressing relations among entities. As a result, the definition of CEs and the use of background knowledge is hard. Logic-based approaches, such as the Event Calculus [Kowalski and Sergot 1986; Artikis et al. 2010a], can naturally and compactly represent relational CE structures. Based on their formal and declarative semantics, they provide solutions that allow one to easily incorporate and exploit background knowledge. In contrast to statistical methods, however, they cannot handle uncertainty which naturally exists in many real-world event recognition applications.

Event recognition systems often have to deal with data that involves a significant amount of uncertainty [Shet et al. 2007; Artikis et al. 2010a; Etzion and Niblett 2010, Section 11.2; Gal et al. 2011]: (a) Low-level detection systems often cannot detect all SDEs required for CE recognition, e.g. due to a limited number of sensing sources. Logical definitions of CEs, therefore, have to be constructed upon a limited and often insufficient dictionary of SDEs. (b) Partial and noisy observations result in incomplete and erroneous SDE streams. For example, a sensor may fail for some period of time and stop sending information, interrupting the detection of a SDE. Similarly, noise in the signal transmission may distort the observed values. (c) When Machine Learning algorithms are used, similar patterns of SDEs may be inconsistently annotated. As a result, CE definitions and background knowledge, either learnt from data or derived by domain experts cannot strictly follow the annotation. Under such situations of uncertainty, the performance of an event recognition system may be seriously compromised.

In the presence of some of the aforementioned types of uncertainty, e.g. partial SDE streams and inconsistent annotations, the CE definitions of a logic-based Event Recognition system cannot capture perfectly the conditions under which a CE occurs. In this work we focus on the recognition of CEs with imperfect definitions, given ambiguously annotated SDEs and incomplete input streams. In particular, we propose a probabilistic version of the Event Calculus that employs Markov Logic Networks (MLNs) [Domingos and Lowd 2009].

The Event Calculus is a formalism for representing events and their effects. Beyond the advantages stemming from the fact that it is a logic-based formalism with clear semantics, one of the most interesting properties of the Event Calculus is that it handles the persistence of CEs with domain-independent axioms. On the other hand, MLNs are a generic statistical relational framework that combines the expressivity of first-order logic with the formal probabilistic properties of undirected graphical models — see de Salvo Braz et al. [2008], Raedt and Kersting [2010] and Blockeel [2011] for surveys on logic-based relational probabilistic models. By combining the Event Calculus

with MLNs, we present a principled and powerful probabilistic logic-based method for event recognition.

In particular the contributions of this work are the following:

- A probabilistic version of the Event Calculus for the task of event recognition. The method inherits the domain-independent properties of the Event Calculus and supports the probabilistic recognition of CEs with imperfect definitions.
- Efficient representation of the Event Calculus axioms and CE definitions in MLNs, in order to avoid the combinatorial explosion caused by the expressivity of the logical formalism.
- The method extends MLNs with a preprocessing step that translates the entire knowledge base into compact form, in order to reduce the complexity of learning and inference.
- A thorough study of the behaviour of CE persistence. Under different conditions of interest, the method can model various types of CE persistence, ranging from deterministic to purely probabilistic.

To demonstrate the benefits of the proposed approach, the method is evaluated in the real-life event recognition task of human activity recognition. The method is compared against its crisp predecessor, as well as a purely statistical model, based on linear-chain Conditional Random Fields. The definitions of CEs are domain-dependent rules that are given by humans and expressed using the language of the Event Calculus. The method processes the rules in the knowledge base and produces Markov networks of manageable size and complexity. Each rule can be associated with a weight value, indicating a degree of confidence in it. Weights are automatically estimated from a training set of examples. The input to the recognition system is a sequence of SDEs expressed as a narrative of ground predicates. Probabilistic inference is used to recognise CEs.

The remainder of the paper is organised as follows. First, in Section 2 we introduce the target activity recognition application. Then, in Section 3 we present the proposed probabilistic Event Calculus. Furthermore, in this section we describe the representational simplifications and transformations that we employ, in order to produce compact knowledge base for efficient inference and learning in Markov Logic Networks. In Section 4, we study the behaviour of the probabilistic formalism. In Section 5, we demonstrate the benefits of probabilistic modeling, through experiments in the real-life activity recognition application. Finally in Sections 6 and 7, we present related work and outline directions for further research.

2. APPLICATION TO ACTIVITY RECOGNITION

In event recognition the input is composed of a linearly ordered stream of time-stamped symbols, called simple, derived events (SDEs). Based on such time-stamped input SDEs, the symbolic event recognition system aims to detect patterns of events. Every recognised instance of an event pattern indicates a composite event (CE). Such patterns are expressed as definitions of CEs and capture knowledge of significant events for the target application. CE definitions are usually given by domain experts or estimated by machine learning techniques. An important characteristic, therefore, is that event recognition targets applications where the input and output data streams are temporally associated and expressed in the form of events. Additionally, all CEs express known types of behaviour.

To demonstrate our method, we apply it to video surveillance in public spaces, using the publicly available benchmark dataset of the CAVIAR project¹. The aim is to

¹<http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1>

recognise activities that take place between multiple persons, by exploiting information about observed activities of individuals. The dataset comprises 28 surveillance videos, where each frame is annotated by human experts from the CAVIAR team on two levels. The first level contains SDEs that concern activities of individual persons or the state of objects. The second level contains CE annotations, describing the activities between multiple persons and/or objects, e.g. people *meeting* and *moving* together, *leaving an object*, etc. In this paper, we focus on the recognition of the CEs *meeting* and *moving*, for which the dataset contains a sufficient amount of training examples. Below we briefly describe the inputs of the activity recognition application:

- (1) The input to our method is a stream of SDEs, representing people *walking*, *running*, *staying active*, or *inactive*. The first and the last time that a person or an object is tracked are represented by the SDEs *enter* and *exit*. The coordinates of tracked persons or objects are also taken into consideration, in order to express qualitative spatial relations, e.g. two persons being relatively close to each other.
- (2) The definitions of the CE *meeting* and *moving* were developed in [Artikis et al. 2010b]. These definitions take the form of common-sense rules and describe the conditions under which a CE starts or ends. For example, when two persons are walking together with the same orientation, then *moving* starts being recognised. Similarly, when the same persons walk away from each other, then *moving* stops being recognised.

Based on the input stream of SDEs and the CE definitions, the aim is to recognise instances of the CEs of interest.

We do not process the raw video data in order to recognise individual activities. Instead we use the SDEs provided in the CAVIAR dataset. Despite the manual annotation, the level of ambiguity in the dataset is high. First, there is a limited dictionary of SDEs and context variables and thus the same pattern of SDEs may be ambiguously annotated by the evaluators [List et al. 2005]. Furthermore, the CE definitions that we use in this work comprise common domain knowledge and do not follow the annotation strictly. As a result, regarding the annotation of the dataset, the CE definitions are imperfect.

3. THE PROBABILISTIC EVENT CALCULUS

The Event Calculus, originally introduced by Kowalski and Sergot [1986], is a many-sorted first-order predicate calculus for reasoning about events and their effects. A number of different dialects have been proposed using either logic programming or classical logic [Shanahan 1999; Miller and Shanahan 2002; Mueller 2008]. Most Event Calculus dialects share the same ontology and core domain-independent axioms. The ontology consists of *time-points*, *events* and *fluents*. The underlying time model is often linear and may represent time-points as real or integer numbers. A *fluent* is a property whose value may change over time. When an *event* occurs it may change the value of a fluent. The core domain-independent axioms define whether a fluent holds or not at a specific time-point. Moreover, the axioms incorporate the common sense *law of inertia*, according to which fluents persist over time, unless they are affected by the occurrence of some event.

The formalism can compactly represent complex event relations, where fluents and events correspond to composite events (CEs) and simple, derived events (SDEs), respectively. Domain-independent axioms and CE definitions can be expressed as a set of first-order logic formulas. Each formula imposes a constraint over the set of possible worlds, that is, Herbrand interpretations. However, the Event Calculus cannot handle uncertainty adequately. A missed or an erroneous SDE detection may have a significant effect on the event recognition results, e.g. causing the erroneous recognition

of a CE. The CE definitions, may not strictly follow the annotation or fail to capture perfectly all the conditions under which a CE occurs.

To deal with uncertainty, the proposed method ($MLN-EC$) combines a discrete variant of the Event Calculus with the probabilistic framework of Markov Logic Networks (MLNs). Figure 1 outlines the structure of the method. The input to $MLN-EC$ is composed of a stream of SDE occurrences and a set of domain-dependent CE definitions. The CE definitions take the form of common-sense rules and describe the conditions under which a CE starts or ends. The method combines the given definitions with the domain-independent axioms of $MLN-EC$, generating a compact knowledge base that forms a template for the production of Markov Networks. The method employs Markov Logic Networks, in order to perform learning and probabilistic inference. Learning takes place once, in order to estimate the parameters of the model from data. Using probabilistic inference, $MLN-EC$ recognises the CE of interest for the given input stream of SDEs.

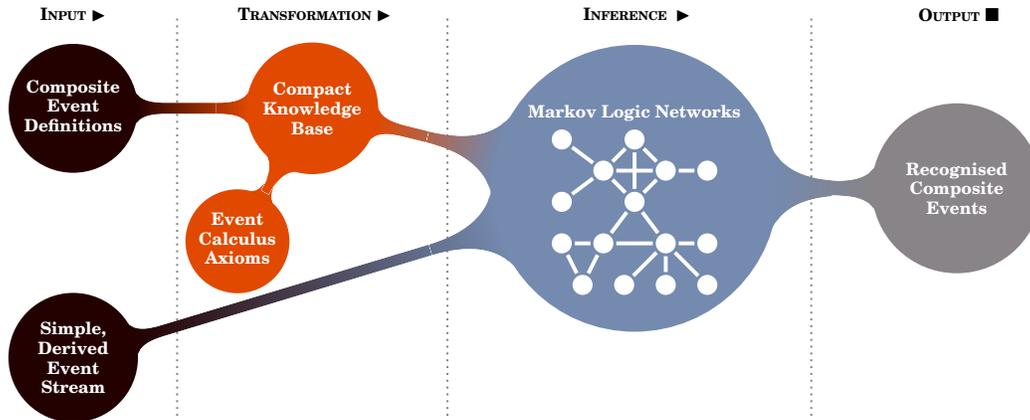


Fig. 1: Event recognition with the $MLN-EC$

The use of MLNs for inference and learning requires the grounding of the entire knowledge base used for event recognition. Unless optimised, this process leads to unmanageably large ground Markov Networks, where inference and learning become practically infeasible. In this work we address this problem at two levels:

- (1) We propose a simplified dialect of the Event Calculus for event recognition. Table I summarises the elements of the $MLN-EC$. Variables (starting with an upper-case letter) are assumed to be universally quantified unless otherwise indicated. Predicates, functions and constants start with a lower-case letter.
- (2) We extend MLNs with a preprocessing step that transforms the knowledge base into a logically stronger one. The simplifications that are made during the preprocessing reduce the number of random variables in the ground Markov network, simplifying the complexity of inference and learning.

In the rest of this section, we present in detail each step of the process illustrated in Figure 1.

3.1. Input to $MLN-EC$

The *happens* predicate (see Table I) expresses the input evidence, determining the occurrence of a SDE at a specific time-point. The input stream of SDEs, therefore,

Table I: The *MLN-EC* predicates

Predicate	Meaning
$happens(E, T)$	Event E occurs at time-point T
$holdsAt(F, T)$	Fluent F holds at time-point T
$initiatedAt(F, T)$	Fluent F is initiated at time-point T
$terminatedAt(F, T)$	Fluent F is terminated at time-point T

is represented in *MLN-EC* as a *narrative* of ground *happens* predicates. The predicates *initiatedAt* and *terminatedAt* specify under which circumstances a fluent — representing a CE — is to be initiated or terminated at a specific time-point. Formally, in *MLN-EC* the domain-dependent rules of the initiation and/or termination of some $fluent_1$ over some domain-specific entities X and Y take the following general form:

$$\begin{aligned}
 &initiatedAt(fluent_1(X, Y), T) \Leftarrow \\
 &\quad happens(event_i(X), T) \wedge \dots \wedge \\
 &\quad holdsAt(fluent_j(X), T) \wedge \dots \wedge \\
 &\quad Conditions[X, Y, T] \\
 &terminatedAt(fluent_1(X, Y), T) \Leftarrow \\
 &\quad happens(event_k(X), T) \wedge \dots \wedge \\
 &\quad holdsAt(fluent_l(X), T) \wedge \dots \wedge \\
 &\quad Conditions[X, Y, T]
 \end{aligned} \tag{1}$$

In this work we consider finite domains of time-points, events and fluents, that are represented by the finite sets \mathcal{T} , \mathcal{E} and \mathcal{F} , respectively. All individual entities that appear in a particular event recognition task, e.g. persons, objects, etc., are represented by the constants of the finite set \mathcal{O} . $Conditions[X, Y, T]$ in (1) is a set of predicates that introduce further constraints in the definition, referring to time $T \in \mathcal{T}$ and entities $X, Y \in \mathcal{O}$. The predicates *happens* and *holdsAt*, as well as those appearing in $Conditions[X, Y, T]$, may also be negated. The initiation and termination of a fluent can be defined by more than one rule, each capturing a different initiation and termination case. With the use of *happens* predicates, we can define a CE over SDE observations. Similarly, with the *holdsAt* predicate we can define a CE over other CE, in order to create hierarchies of CE definitions. In both *initiatedAt* and *terminatedAt* rules, the use of *happens*, *holdsAt* and $Conditions[X, Y, T]$ is optional and varies according to the requirements of the target event recognition application.

The input stream of SDEs to *MLN-EC* is represented by a narrative of ground *happens* predicates, e.g. in activity recognition *happens* predicates represent people *walking*, *running*, staying *active*, *inactive*, etc. The coordinates of tracked persons or objects are preprocessed and represented by the utility predicates *close* and *orientationMove*, expressing qualitative spatial relations. As an example, consider the following fragment of a narrative:

$$\begin{aligned}
 &\dots \\
 &\quad happens(walking(id_1), 100) \\
 &\quad happens(walking(id_2), 100) \\
 &\quad orientationMove(id_1, id_2, 100) \\
 &\quad close(id_1, id_2, 34, 100) \\
 &\dots \\
 &\quad happens(active(id_1), 200) \\
 &\quad happens(active(id_2), 200) \\
 &\dots
 \end{aligned} \tag{2}$$

According to the above narrative, it has been observed that at time-point 100 the persons id_1 and id_2 are *walking* close to each other with similar orientation. The ground predicate *close* indicates that the distance between the two persons at time-point 100 is below the specified threshold of 34 pixels. Similarly, *orientationMove* states that their orientation is almost the same (e.g. the difference is below 45 degrees). Later at time-point 200 both persons are staying active, e.g. they are moving their arms, while staying at the same position.

The input CE definitions of the *meeting* and *moving* activities were developed in [Artikis et al. 2010b]. These definitions are common-sense rules that are expressed using the general form (1). Consider the following definition of the *meeting* CE in our example.

$$\begin{aligned} \textit{initiatedAt}(\textit{meeting}(ID_1, ID_2), T) \Leftarrow \\ \textit{happens}(\textit{active}(ID_1), T) \wedge \\ \neg \textit{happens}(\textit{running}(ID_2), T) \wedge \\ \textit{close}(ID_1, ID_2, 25, T) \end{aligned} \quad (3)$$

$$\begin{aligned} \textit{initiatedAt}(\textit{meeting}(ID_1, ID_2), T) \Leftarrow \\ \textit{happens}(\textit{inactive}(ID_1), T) \wedge \\ \neg \textit{happens}(\textit{running}(ID_2), T) \wedge \\ \neg \textit{happens}(\textit{active}(ID_2), T) \wedge \\ \textit{close}(ID_1, ID_2, 25, T) \end{aligned} \quad (4)$$

$$\begin{aligned} \textit{terminatedAt}(\textit{meeting}(ID_1, ID_2), T) \Leftarrow \\ \textit{happens}(\textit{walking}(ID_1), T) \wedge \\ \neg \textit{close}(ID_1, ID_2, 34, T) \end{aligned} \quad (5)$$

$$\begin{aligned} \textit{terminatedAt}(\textit{meeting}(ID_1, ID_2), T) \Leftarrow \\ \textit{happens}(\textit{running}(ID_1), T) \end{aligned} \quad (6)$$

$$\begin{aligned} \textit{terminatedAt}(\textit{meeting}(ID_1, ID_2), T) \Leftarrow \\ \textit{happens}(\textit{exit}(ID_1), T) \end{aligned} \quad (7)$$

According to rules (3) and (4), the *meeting* CE is initiated when the people involved interact with each other, i.e. at least one of them is active or inactive, the other is not running, and the measured distance between them is at most 25 pixels. The *meeting* CE is terminated when people walk away from each other (rule 5), when one of them is running (rule 6), or has exited the scene (rule 7).

The definition of the CE that people are *moving* together is represented as follows:

$$\begin{aligned} \textit{initiatedAt}(\textit{moving}(ID_1, ID_2), T) \Leftarrow \\ \textit{happens}(\textit{walking}(ID_1), T) \wedge \\ \textit{happens}(\textit{walking}(ID_2), T) \wedge \\ \textit{orientationMove}(ID_1, ID_2, T) \wedge \\ \textit{close}(ID_1, ID_2, 34, T) \end{aligned} \quad (8)$$

$$\begin{aligned} \textit{terminatedAt}(\textit{moving}(ID_1, ID_2), T) \Leftarrow \\ \textit{happens}(\textit{walking}(ID_1), T) \wedge \\ \neg \textit{close}(ID_1, ID_2, 34, T) \end{aligned} \quad (9)$$

$$\begin{aligned} \textit{terminatedAt}(\textit{moving}(ID_1, ID_2), T) \Leftarrow \\ \textit{happens}(\textit{active}(ID_1), T) \wedge \\ \textit{happens}(\textit{active}(ID_2), T) \end{aligned} \quad (10)$$

$$\begin{aligned} \textit{terminatedAt}(\textit{moving}(ID_1, ID_2), T) \Leftarrow \\ \textit{happens}(\textit{active}(ID_1), T) \wedge \\ \textit{happens}(\textit{inactive}(ID_2), T) \end{aligned} \quad (11)$$

$$\begin{aligned} \text{terminatedAt}(\text{moving}(ID_1, ID_2), T) \Leftarrow \\ \text{happens}(\text{running}(ID_1), T) \end{aligned} \quad (12)$$

$$\begin{aligned} \text{terminatedAt}(\text{moving}(ID_1, ID_2), T) \Leftarrow \\ \text{happens}(\text{exit}(ID_1), T) \end{aligned} \quad (13)$$

According to rule (8), the *moving* CE is initiated when two persons ID_1 and ID_2 are walking close to each other (their distance is at most 34 pixels) with almost the same orientation. The *moving* CE is terminated under several cases: (a) When people walk away from each other, i.e. they have a distance larger than 34 pixels, (b) when they stop moving, i.e. either both are active, or one is active while the other is inactive, represented by rules (10) and (11). (c) When one of them is running or exiting the scene, represented by rules (12) and (13), respectively.

3.2. The Event Calculus Axioms

We propose a simplified dialect of the discrete version of the Event Calculus. The representation of the formalism reduces the complexity of the produced ground Markov Networks for probabilistic inference and learning. *MLN-EC* is based on the Discrete Event Calculus (DEC), which has been proven to be logically equivalent to the Event Calculus when the domain of time-points is limited to integers [Mueller 2008]. Compared to DEC, *MLN-EC* is designed for the task of event recognition. For that reason, *MLN-EC* employs only domain-independent axioms that determine the influence of events to fluents and the inertia of fluents. We do not consider the predicates and axioms stating when a fluent is not subject to inertia (*releases* and *releasedAt*), as well as its discrete change based on some domain-specific mathematical function (*trajectory* and *antiTrajectory*). Furthermore, the initiation and termination of fluents are defined only using the general form (1) — as presented in Section 3.1.

The *MLN-EC* axioms that determine when a fluent holds are defined as follows:

$$\begin{aligned} \text{holdsAt}(F, T+1) \Leftarrow \\ \text{initiatedAt}(F, T) \end{aligned} \quad (14)$$

$$\begin{aligned} \text{holdsAt}(F, T+1) \Leftarrow \\ \text{holdsAt}(F, T) \wedge \\ \neg \text{terminatedAt}(F, T) \end{aligned} \quad (15)$$

Axiom (14) defines that if a fluent F is initiated at time T , then it holds at the next time-point. Axiom (15) specifies that a fluent continues to hold unless it is terminated.

The axioms that determine when a fluent does not hold are defined similarly:

$$\begin{aligned} \neg \text{holdsAt}(F, T+1) \Leftarrow \\ \text{terminatedAt}(F, T) \end{aligned} \quad (16)$$

$$\begin{aligned} \neg \text{holdsAt}(F, T+1) \Leftarrow \\ \neg \text{holdsAt}(F, T) \wedge \\ \neg \text{initiatedAt}(F, T) \end{aligned} \quad (17)$$

According to axiom (16), if a fluent F is terminated at time T then it does not hold at the next time-point. Axiom (17) states that a fluent continues not to hold unless it is initiated. Axioms (15) and (17) express the inertia of fluents.

The choice of the Event Calculus dialect has a significant impact on the grounding process. For example, Shanahan's Full Event Calculus [Shanahan 1999] employs axioms that contain triply quantified time-point variables. As a result, the number of their groundings has a cubic relation to the number of time-points. Furthermore, that formalism contains existentially quantified variables over events and time-points. Dur-

ing MLN grounding existentially quantified formulas are replaced by the disjunction of their groundings [Domingos and Lowd 2009]. This leads to a large number of disjunctions and a combinatorial explosion of the number of clauses, producing unmanageably large Markov networks.

In contrast, the proposed *MLN-EC* is based on the DEC, where the domain-independent axioms are defined over successive time-points. For example, axiom (14) produces one clause in Conjunctive Normal Form and has two distinct variables F and T . Therefore, the number of its groundings is determined by the Cartesian product of the corresponding variable-binding constraints, that is $|\mathcal{F}| \times |\mathcal{T}|$. Assuming that the domain of fluents \mathcal{F} is relatively small compared to the domain of time-points \mathcal{T} , the number of groundings of axiom (14) grows linearly to the number of time-points. In contrast to DEC, *initiatedAt* and *terminatedAt* are defined only in terms of fluents and time-points. This representation reduces further the number of variables and eliminates the existential quantification in the domain-independent axioms. Consequently, *MLN-EC* produces a substantially smaller number of ground clauses, than many other dialects of the Event Calculus.

3.3. Compact Knowledge Base

The proposed simplified discrete Event Calculus dialect of the previous section makes the number of ground clauses linearly dependent on the number of time-points. In addition, we can achieve significant improvements in the size of the ground Markov Networks, by taking advantage of the Closed World Assumption. In *MLN-EC* we extend MLNs with a preprocessing procedure that simplifies the knowledge base of CE definitions (e.g. rules (3)–(13)) and Event Calculus axioms (i.e. rules (14)–(17)).

The rules in the form of (1) describe explicitly the conditions in which fluents are initiated or terminated. It is usually impractical to define also when a fluent is *not* initiated and *not* terminated. However, the open-world semantics of first-order logic result in an inherent uncertainty about the value of a fluent for many time-points. In other words, if at a specific time-point no event that terminates or initiates a fluent happens, we cannot rule out the possibility that the fluent has been initiated or terminated. As a result, we cannot determine whether a fluent holds or not, leading to the loss of inertia.

This is a variant of the well-known frame problem and one solution for the Event Calculus in first-order logic is the use of circumscription [McCarthy 1980; Lifschitz 1994; Shanahan 1997; Doherty et al. 1997; Mueller 2008]. The aim of circumscription is to automatically rule out all those conditions which are not explicitly entailed by the given formulas. Hence, circumscription introduces a closed-world assumption to first-order logic.

Technically, we perform circumscription by predicate completion — a syntactic transformation where formulas are translated into logically stronger ones. Similar to Mueller [2008], we perform predicate completion on *initiatedAt* and *terminatedAt* predicates. Due to the form of CE definitions (see formalisation (1)), the result of predicate completion in *MLN-EC* concerns each CE separately, e.g. *initiatedAt(meeting(ID₁, ID₂), T)*, rather than a generic *initiatedAt(F, T)* predicate. Thereafter, we eliminate the *initiatedAt* and *terminatedAt* predicates from the knowledge base, by exploiting the equivalences resulting from predicate completion. In cases where the definitions of the initiation or termination of a specific CE are missing, the corresponding initiation or termination is considered *False* for all time-points, e.g. *terminatedAt(fluent(X, Y), T) ⇔ False*.

To illustrate the form of the resulting knowledge base, consider the domain-dependent definition of *meeting* — i.e. rules (3)–(7). After predicate completion, these

rules will be replaced by the following formulas:

$$\begin{aligned}
 \textit{initiatedAt}(\textit{meeting}(ID_1, ID_2), T) \Leftrightarrow & \\
 & (\textit{happens}(\textit{active}(ID_1), T) \wedge \\
 & \neg \textit{happens}(\textit{running}(ID_2), T) \wedge \\
 & \textit{close}(ID_1, ID_2, 25, T)) \vee \\
 & (\textit{happens}(\textit{inactive}(ID_1), T) \wedge \\
 & \neg \textit{happens}(\textit{running}(ID_2), T) \wedge \\
 & \neg \textit{happens}(\textit{active}(ID_2), T) \wedge \\
 & \textit{close}(ID_1, ID_2, 25, T))
 \end{aligned} \tag{18}$$

$$\begin{aligned}
 \textit{terminatedAt}(\textit{meeting}(ID_1, ID_2), T) \Leftrightarrow & \\
 & (\textit{happens}(\textit{walking}(ID_1), T) \wedge \\
 & \neg \textit{close}(ID_1, ID_2, 25, T)) \vee \\
 & \textit{happens}(\textit{running}(ID_1), T) \vee \\
 & \textit{happens}(\textit{exit}(ID_1), T)
 \end{aligned} \tag{19}$$

The resulting rules (18) and (19) define all conditions under which the *meeting* CE is initiated or terminated. Any other event occurrence cannot affect this CE, as it cannot initiate the CE or terminate it. Based on the equivalence in formula (18), the domain-independent axiom (14) is re-written into the following specialised form²:

$$\begin{aligned}
 \textit{holdsAt}(\textit{meeting}(ID_1, ID_2), T+1) \Leftarrow & \\
 & \textit{happens}(\textit{active}(ID_1), T) \wedge \\
 & \neg \textit{happens}(\textit{running}(ID_2), T) \wedge \\
 & \textit{close}(ID_1, ID_2, 25, T)
 \end{aligned} \tag{20}$$

$$\begin{aligned}
 \textit{holdsAt}(\textit{meeting}(ID_1, ID_2), T+1) \Leftarrow & \\
 & \textit{happens}(\textit{inactive}(ID_1), T) \wedge \\
 & \neg \textit{happens}(\textit{running}(ID_2), T) \wedge \\
 & \neg \textit{happens}(\textit{active}(ID_2), T) \wedge \\
 & \textit{close}(ID_1, ID_2, 25, T)
 \end{aligned}$$

Similarly, the inertia axiom (15) can be re-written according to (19) as follows:

$$\begin{aligned}
 \textit{holdsAt}(\textit{meeting}(ID_1, ID_2), T+1) \Leftarrow & \\
 & \textit{holdsAt}(\textit{meeting}(ID_1, ID_2), T) \wedge \\
 & \neg \left((\textit{happens}(\textit{walking}(ID_1), T) \wedge \right. \\
 & \quad \left. \neg \textit{close}(ID_1, ID_2, 25, T)) \vee \right. \\
 & \quad \left. \textit{happens}(\textit{running}(ID_1), T) \vee \right. \\
 & \quad \left. \textit{happens}(\textit{exit}(ID_1), T) \right)
 \end{aligned} \tag{21}$$

The result of this transformation procedure replaces the original set of domain-independent axioms and domain-dependent CE definitions with a logically stronger knowledge base. The rules in the resulting knowledge base form the template that MLNs will use to produce ground Markov networks. The transformed formulas produce considerably more compact ground Markov networks than the original ones, as the clauses to be grounded are reduced. Moreover, the predicates *initiatedAt* and

²This direct re-writing of rule (18) results to a single formula that contains the disjunction of formula (18). However, for reasons that have to do with the handling of uncertainty in MLNs and will be discussed in a later section, in rule (20) we choose to equivalently represent it using two separate formulas.

terminatedAt are eliminated and the corresponding random variables are not added to the network. This reduction decreases substantially the space of possible worlds, since the target random variables of the network are limited only to the corresponding *holdsAt* ground predicates. Specifically, the space of possible worlds is reduced from $2^{3 \times |\mathcal{F}| \times |\mathcal{T}|}$ to $2^{|\mathcal{F}| \times |\mathcal{T}|}$ — where $|\mathcal{T}|$ and $|\mathcal{F}|$ denote the number of distinct time-points and fluents, respectively. These reductions improve the computational performance of the probabilistic inference. Furthermore, due to the reduced space of possible worlds, the same number of sampling iterations results in better probability estimates.

Formally, the resulting knowledge base is composed of rules having the following form:

$$\Sigma = \begin{cases} \text{holdsAt}(\text{fluent}_1(X, Y), T+1) \Leftarrow \\ \text{happens}(\text{event}_i(X), T) \wedge \dots \wedge \text{Conditions}[X, Y, T] & (22) \\ \dots \\ \neg \text{holdsAt}(\text{fluent}_1(X, Y), T+1) \Leftarrow \\ \text{happens}(\text{event}_j(X), T) \wedge \dots \wedge \text{Conditions}[X, Y, T] & (23) \\ \dots \end{cases}$$

$$\Sigma' = \begin{cases} \text{holdsAt}(\text{fluent}_1(X, Y), T+1) \Leftarrow \\ \text{holdsAt}(\text{fluent}_1(X, Y), T) \wedge \\ \neg((\text{happens}(\text{event}_j(X), T) \wedge \dots \wedge \text{Conditions}[X, Y, T]) \vee \dots) & (24) \\ \dots \\ \neg \text{holdsAt}(\text{fluent}_1(X, Y), T+1) \Leftarrow \\ \neg \text{holdsAt}(\text{fluent}_1(X, Y), T) \wedge \\ \neg((\text{happens}(\text{event}_i(X), T) \wedge \dots \wedge \text{Conditions}[X, Y, T]) \vee \dots) & (25) \\ \dots \end{cases}$$

The rules in (22)–(25) can be separated into two subsets. The former set Σ contains specialised definitions of axioms (14) and (16), specifying that a fluent holds (or does not hold) when its initiation (or termination) conditions are met. The latter set Σ' contains specialised definitions of the inertia axioms (15) and (17), specifying whether a specific fluent continues to hold or not at any instance of time.

The presented knowledge transformation procedure reduces the size of the produced network, based only on the rules of the knowledge base. This transformation is a pre-processing step of the standard grounding procedure of MLNs.

3.4. Probabilistic Representation based on Markov Logic Networks

For the probabilistic representation of the compact knowledge base of rules in the form of (22)–(25), *MLN-EC* employs the statistical relational learning framework of MLNs³. Each formula F_i in the knowledge base can be associated with a weight value $w_i \in \mathbb{R}$. The higher the value of weight w_i , the stronger the constraint represented by formula F_i . In contrast to classical logic, all worlds (i.e. Herbrand interpretations) become possible with a certain probability. The main idea behind this is that the probability of a world increases as the number of formulas it violates decreases. The knowledge base may contain both hard and soft-constrained formulas. Hard-constrained formulas are associated with an infinite weight value and capture the knowledge which is assumed

³Systems implementing MLN reasoning and learning algorithms can be found at the following addresses:
<http://alchemy.cs.washington.edu>
<http://research.cs.wisc.edu/hazy/tuffy>
<http://code.google.com/p/thebeast>
<http://ias.cs.tum.edu/probcog-wiki>
<https://github.com/anskarl/LoMRF>

to be certain. Therefore, an acceptable world must at least satisfy the hard constraints. Soft constraints capture imperfect knowledge in the domain, allowing for the existence of worlds in which this knowledge is violated.

Formally, the *MLN-EC* knowledge base L of weighted formulas in the form of (22)–(25), together with a finite domain of constants \mathcal{C} , is transformed into a ground Markov network $M_{L,\mathcal{C}}$. All formulas are converted into *clausal form* and each clause is ground according to the domain of its distinct variables. The nodes in $M_{L,\mathcal{C}}$ are Boolean random variables, each one corresponding to a possible grounding of a predicate that appears in L . The predicates of a ground clause form a clique in $M_{L,\mathcal{C}}$. Each clique is associated with a corresponding weight w_i and a Boolean *feature*, taking the value 1 when the ground clause is true and 0 otherwise. The ground $M_{L,\mathcal{C}}$ defines a probability distribution over possible worlds and is represented as a log-linear model.

In event recognition we aim to recognise CEs of interest given the observed streams of SDEs. For this reason we focus on discriminative MLNs [Singla and Domingos 2005], that are akin to Conditional Random Fields [Lafferty et al. 2001; Sutton and McCallum 2007]. Specifically, the set of random variables in $M_{L,\mathcal{C}}$ can be partitioned into two subsets. The former is the set of evidence random variables X , formed by a narrative of input ground *happens* predicates, i.e. the entire sequence of observed SDEs. The latter is the set of random variables Y that correspond to groundings of query *holdsAt* predicates. The joint probability distribution of a possible assignment of $Y=\mathbf{y}$, conditioned over a given assignment of $X=\mathbf{x}$, is defined as follows:

$$P(Y=\mathbf{y} | X=\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp \left(\sum_{i=1}^{|\mathcal{F}_c|} w_i n_i(\mathbf{x}, \mathbf{y}) \right) \quad (26)$$

The vectors $\mathbf{x} \in \mathcal{X}$ and $\mathbf{y} \in \mathcal{Y}$ represent a possible assignment of evidence X and query/hidden variables Y , respectively. \mathcal{X} and \mathcal{Y} are the sets of possible assignments that the evidence X and query/hidden variables Y can take. \mathcal{F}_c is the set of clauses produced from the knowledge base L and the domain of constants $\mathcal{C}=\mathcal{T} \cup \mathcal{O} \cup \mathcal{E} \cup \mathcal{F}$. The scalar value w_i is the weight of the i -th clause and $n_i(\mathbf{x}, \mathbf{y})$ is the number of satisfied groundings of the i -th clause in \mathbf{x} and \mathbf{y} . $Z(\mathbf{x})$ is the partition function, that normalises over all possible assignments $\mathbf{y}' \in \mathcal{Y}$ of query/hidden variables given the assignment \mathbf{x} , that is, $Z(\mathbf{x}) = \sum_{\mathbf{y}' \in \mathcal{Y}} \exp(\sum_{i=1}^{|\mathcal{F}_c|} w_i n_i(\mathbf{x}, \mathbf{y}'))$.

Equation (26) represents a single exponential model for the joint probability of the entire set of query variables that is globally conditioned on a set of observables. Such a conditional model can have a much simpler structure than a full joint model, e.g. a Bayesian Network. By modelling the conditional distribution directly, the model is not affected by potential dependencies between the variables in X and can ignore them. The model also makes independence assumptions among the random variables Y , and defines by its structure the dependencies of Y on X . Furthermore, conditioning on a specific assignment \mathbf{x} , given by the observed SDEs, reduces significantly the number of possible worlds and inference becomes much more efficient [Singla and Domingos 2005; Minka 2005; Sutton and McCallum 2007].

3.4.1. Inference. Still, directly computing equation (26) is intractable, because the value of $Z(\mathbf{x})$ depends on the relationship among all clauses in the knowledge base. For this reason, a variety of efficient inference algorithms have been proposed in the literature, based on local search and sampling [Poon and Domingos 2006; Singla and Domingos 2006; Biba et al. 2011], variants of Belief Propagation [Singla and Domingos 2008; Kersting et al. 2009], Integer Linear Programming [Riedel 2008; Huynh and Mooney 2009], etc.

In this work we consider two types of inference, i.e. marginal inference and maximum a-posteriori inference (MAP). The former type of inference computes the conditional probability that CEs hold given a narrative of observed SDEs, i.e. $P(\text{holdsAt}(CE, T) = \text{True} \mid \text{SDEs})$. In other words, this probability value measures the confidence that the CE is recognised. Since it is #P-complete to compute this probability, we employ the state-of-the-art sampling algorithm *MC-SAT* [Poon and Domingos 2006] to approximate it. The algorithm combines Markov Chain Monte Carlo sampling with satisfiability testing and even in large state spaces with deterministic dependencies (e.g. hard-constrained formulas) it can approximate this probability efficiently. The latter type of inference, maximum a-posteriori inference, identifies the most probable assignment among all *holdsAt* instantiations that are consistent with the given narrative of observed SDEs, i.e. $\underset{\text{holdsAt}}{\text{argmax}} P(\text{holdsAt}(CE, T) \mid \text{SDEs})$. In MLNs this task reduces

to finding the truth assignment of all *holdsAt* instantiations that maximises the sum of weights of satisfied ground clauses. This is equivalent to the weighted maximum satisfiability problem. The problem is NP-hard in general and in order to find an approximate solution efficiently we employ the LP-relaxed Integer Linear Programming method proposed by Huynh and Mooney [2009].

3.4.2. Weight Learning. The weights of the soft-constrained clauses in MLNs can be estimated from training data, using supervised learning techniques. When the goal is to learn a model that recognises CEs with some confidence (i.e. probability), then the most widely adopted approach is to minimise the negative Conditional Log-Likelihood (CLL) function — derived from equation (26). This can be achieved by using either *first-order* or *second-order* optimisation methods [Singla and Domingos 2005; Lowd and Domingos 2007]. First-order methods apply standard gradient descent optimisation techniques, e.g. the voted perceptron algorithm [Collins 2002; Singla and Domingos 2005], while second-order methods pick a search direction based on the quadratic approximation of the target function. As stated by Lowd and Domingos [2007], second-order methods are more appropriate for MLN training, as they do not suffer from the problem of *ill-conditioning*. In a training set some clauses may have a significantly greater number of satisfied groundings than others, causing the variance of their counts to be correspondingly larger. This situation causes the standard gradient descent methods to converge very slowly, since there is no single appropriate learning rate for all soft-constrained clauses. An alternative approach to CLL function optimisation is max-margin training, which is better suited to problems where the goal is to maximise the classification accuracy [Huynh and Mooney 2009; 2011]. Instead of optimising the CLL function, max-margin training aims to maximise the ratio between the probability of the correct truth assignment of CEs to hold and the closest competing incorrect truth assignment. In this work we assess both the second-order Diagonal Newton algorithm [Singla and Domingos 2005] and the max-margin method proposed by Huynh and Mooney [2009].

3.5. Output of *MLN-EC*

The output of *MLN-EC* is a stream of recognised CEs (activities in our running example). The clauses in the form of (22)–(25) are associated with weights. Given a narrative of SDEs, *MLN-EC* probabilistically determines whether a CE holds or not at any time-point. For example, the SDEs and spatial constraints in narrative (2) at time-points 100 and 200 satisfy the conditions of rules (8) and (10), respectively. Therefore, at time-point 100 the *moving* CE is initiated, increasing the probability of *moving* to hold at the next time-point, i.e. $\text{holdsAt}(\text{moving}(id_1, id_2), 101)$. On the other hand, at time-point 200 the *moving* CE is terminated and this situation reduces the probability

of *moving* to hold at the next time-point. In the following section we give a detailed description of how weights are defined among clauses and how they affect the probability of the recognised CEs.

4. THE BEHAVIOUR OF THE PROBABILISTIC EVENT CALCULUS

Recall that the compact knowledge base of *MLN-EC* is separated into the Σ and Σ' subsets (see Section 3.3). The set Σ is composed of the formulas that specify when a fluent holds (or does not hold), while Σ' is composed of the formulas that specify when a fluent continues to hold or not. As mentioned in Section 3.4, by associating weights to such formulas we define soft constraints, allowing some worlds that do not satisfy these formulas to become likely. For example, consider a knowledge base of Event Calculus axioms and CE definitions (e.g. *meeting* and *moving*) compiled in the form of rules (22)–(25). Given a narrative of SDEs, the probability of a CE to hold at a specific time-point is determined by the probabilities of the worlds in which this CE holds. Each world, in turn, has some probability which is proportional to the sum of the weights of the ground clauses that it satisfies. Consequently, the probability of a CE to hold at a specific instance of time depends on the corresponding constraints of the ground Markov network. Thus, by treating the rules in the Σ and Σ' sets as either hard or soft constraints, we can modify the behaviour of the Event Calculus. Compared to crisp implementations of the Event Calculus, this flexibility of changing the behaviour of the formalism is the main advantage of the proposed method.

4.1. Soft-constrained rules in Σ

In order to illustrate how the probability of a CE is affected when its initiation or termination conditions are met, consider the case that the rules in Σ are soft-constrained while the inertia rules in Σ' remain hard-constrained. By soft-constraining the rules in Σ , the worlds violating their clauses become probable. This situation reduces the certainty with which a CE is recognised when its initiation or termination conditions are met. For example, assume that the initiation rules (20) of the *meeting* CE are associated with weights. As a result, the *meeting* activity is initiated with some certainty, causing the CE to hold with some probability. Depending on the strength of the weights, the worlds that violate these rules become more or less likely. Thus, we can control the level of certainty with which a CE holds or not under the same conditions.

When the initiation conditions are met, the probability of the CE to hold increases. Equivalently, when the termination conditions are satisfied, the probability of the CE decreases. At the same time, all worlds that do not fully satisfy hard-constrained inertia rules in Σ' are rejected, thus the inertia is retained deterministically as in crisp logic.

Figure 2 illustrates this behaviour with the fluent *meeting* that initially does not hold at time 0. According to the narrative of SDEs, the *meeting* activity is initiated at time-points 3 and 10, e.g. satisfying the constraints imposed by rules (3) and (4) respectively. At time 20, the *meeting* activity is terminated by the conditions of rule (5). In crisp Event Calculus, denoted as EC_{crisp} , after its first initiation the *meeting* activity holds with absolute certainty. The second initiation at time 10 does not cause any change and the CE continues to hold. The termination at time 20 causes the CE to not hold, again with absolute certainty, for the remaining time-points. In $MLN-EC_{HI}$ (hard-constrained inertia rules), however, the rules in Σ are soft-constrained. As a result, at time-point 4 the probability of *meeting* to hold increases to some value. Similar to EC_{crisp} , the inertia is fully retained and the probability of *meeting* deterministically persists in the interval 4 to 10. In contrast to EC_{crisp} , the second initiation at time-point 10 increases the certainty of *meeting* to hold. As a result, the probability of *meeting* is

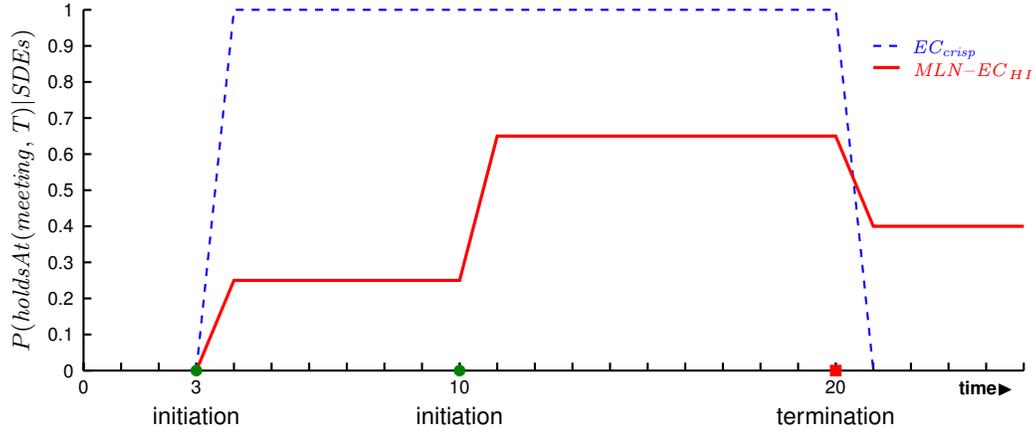


Fig. 2: The probability of the *meeting* CE given some SDE narrative. EC_{crisp} is a crisp Event Calculus. $MLN-EC_{HI}$ is a probabilistic Event Calculus where rules in Σ are soft-constrained, while the inertia rules in Σ' remain hard-constrained.

higher in the interval 11 to 20. In the same manner, the termination at 20 reduces the probability of *meeting* and the CE continues to hold with some reduced probability.

4.2. Soft-constrained inertia rules in Σ'

To illustrate how the behaviour of inertia is affected by soft-constraining the corresponding rules in Σ' , consider that the rules in Σ are hard-constrained. Consequently, when the initiation (or termination) conditions are met, a CE holds (or does not hold) with absolute certainty. The persistence of a CE depends on its inertia rules in Σ' . If the inertia of *holdsAt* is hard-constrained, the worlds in which an initiated CE does not hold are rejected. Similarly, by keeping the inertia of \neg *holdsAt* hard-constrained, all worlds in which a terminated CE holds are rejected. By soft-constraining these rules we control the strength of the inertia constraints. Event when the inertia constraints are partially violated, the probability of the CE is affected. Thus, the persistence of *holdsAt* and \neg *holdsAt* is gradually lost over successive time-points. When allowing the constraints of *holdsAt* inertia to be violated, the probability of a CE gradually drops. Similarly, by allowing the constraints representing the inertia of \neg *holdsAt* to be violated, the probability of a CE gradually increases. The lower the value of the weight on the constraint, the more probable the worlds that violate the constraints become. In other words, weight values in Σ' cause CE to persist for longer or shorter time periods.

Since the sum of the probabilities of *holdsAt* and \neg *holdsAt* for a specific CE is always equal to 1, the relative strength of *holdsAt* and \neg *holdsAt* rules in Σ' determines the type of inertia in the model. The following two general cases can be distinguished.

Equally strong inertia constraints. All rules in Σ' are equally soft-constrained, i.e. they are associated with the same weight value. Consequently, both inertia rules of *holdsAt* and \neg *holdsAt* for a particular CE impose constraints of equal importance, allowing worlds that violate them to become likely. As a result, in the absence of useful evidence, the probability of *holdsAt* will tend to approximate the value 0.5. For example, Figure 3(a) illustrates soft persistence for the *meeting* CE when it holds with absolute certainty at time-point 0, and thereafter nothing happens to initiate or terminate it. The curve $MLN-EC_{Steq}$ (soft-constrained inertia rules with equal weights) shows the behaviour of inertia in this case. As time evolves, the probability of *meeting* appears to

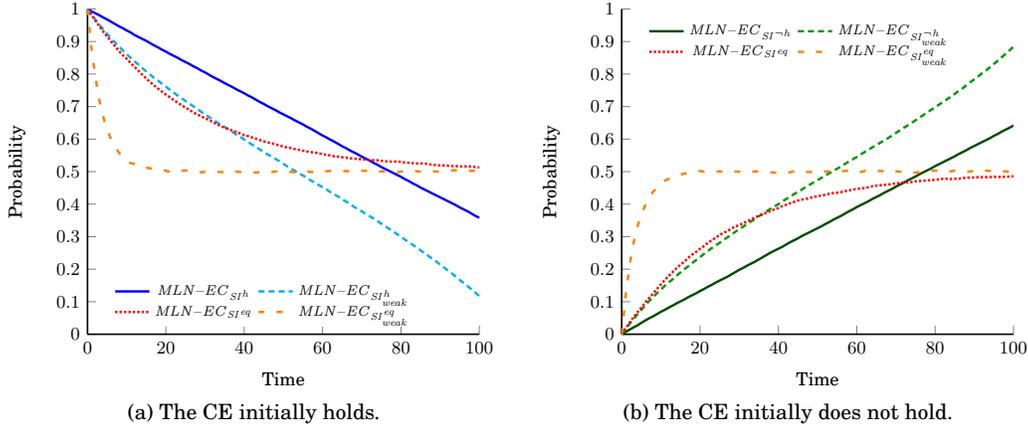


Fig. 3: In both figures SDEs occur that lead to the partial satisfaction of the initiation/termination conditions of a CE in the interval 0 to 100. In the left figure the CE holds at time 0 with absolute certainty, while in the right figure the CE does not hold at time 0.

gradually drop, converging to 0.5. If we assign weaker weights to the inertia axioms, shown by the $MLN-EC_{SI^{weak}}$ curve, the probability of *meeting* drops more sharply. Similarly, in Figure 3(b), the *meeting* CE is assumed to not hold initially. As time evolves, the probability of *meeting* gradually increases up to the value 0.5, as shown by the $MLN-EC_{SI^{eq}}$ and $MLN-EC_{SI^{weak}}$ curves respectively.

Inertia constraints of different strength. When the inertia rules of *holdsAt* and \neg *holdsAt* for a particular CE in Σ' have different weights, the probability of the CE will no longer converge to 0.5. Since the weights impose constraints with different confidence, worlds violating the stronger constraints become less likely than worlds violating the weaker ones. Depending on the relative strength of the weights, the probability of the CE may converge either to 1.0 or 0.0. The relative strength of the weights affects also the rate at which the probability of CE changes. As an extreme example, in Figure 3(a), the rules for the inertia of \neg *holdsAt* remain hard-constrained. By assigning weights to the rules for the inertia of *holdsAt*, the persistence of the CE is lost. Since the inertia constraints of *holdsAt* are weaker than the constraints of \neg *holdsAt*, worlds violating the former set of constraints will always be more likely. As a result, the probability of the CE will continue to drop, even below 0.5. The curves $MLN-EC_{SI^h}$ (soft-constrained inertia of *holdsAt*) and $MLN-EC_{SI^{weak}}$ (weaker *holdsAt* inertia constraints) illustrate how the probability of *meeting* drops sharply towards 0.0. The weaker the constraints ($MLN-EC_{SI^{weak}}$) the steeper the drop. In a similar manner, when the inertia constraints of \neg *holdsAt* are weaker than the constraints of *holdsAt*, the probability of CE gradually increases and may reach values above 0.5 — presented by the $MLN-EC_{SI^{-h}}$ (soft-constrained inertia of \neg *holdsAt*) and $MLN-EC_{SI^{weak}}$ (weaker \neg *holdsAt* inertia constraints) cases in Figure 3(b).

As explained in Section 3.3, the inertia rule of a CE may consist of a large body of conditions, e.g. rule (21). Depending on the number of conditions involved, the rule may be decomposed into several clauses, each corresponding to a different subset of conditions. For instance, the following two clauses are added to Σ' by the inertia rule

(21):

$$\begin{aligned} & \text{happens}(\text{walking}(ID_1), T) \vee \text{happens}(\text{running}(ID_1), T) \vee \text{happens}(\text{exit}(ID_1), T) \vee \\ & \neg \text{holdsAt}(\text{meeting}(ID_1, ID_2), T) \vee \text{holdsAt}(\text{meeting}(ID_1, ID_2), T+1) \end{aligned} \quad (27)$$

$$\begin{aligned} & \neg \text{close}(ID_1, ID_2, 25, T) \vee \text{happens}(\text{running}(ID_1), T) \vee \text{happens}(\text{exit}(ID_1), T) \vee \\ & \neg \text{holdsAt}(\text{meeting}(ID_1, ID_2), T) \vee \text{holdsAt}(\text{meeting}(ID_1, ID_2), T+1) \end{aligned} \quad (28)$$

The above clauses contain literals from the termination rules of the *meeting* CE. Often, some of these clauses become trivially satisfied. For example, at time-point 10 both persons ID_1 and ID_2 are *active*, while their distance is above the threshold of 25 pixels, i.e. $\text{close}(ID_1, ID_2, 25, 10) = \text{False}$. Consequently, the grounding of clause (28) at time-point 10 is trivially satisfied for all possible worlds. Although the *meeting* CE is not terminated at time-point 10, because clause (27) is not satisfied, the satisfaction of clause (28) reduces the probability of *holdsAt* for the CE. This is because the inertia at time-point 10 is now supported only by the satisfaction of the ground clause (27). In other words, the difference between the probabilities of worlds that violate the inertia of *holdsAt* and worlds that do not, is reduced.

To illustrate this phenomenon, consider the example cases in Figure 4(a) where only the rules about the inertia of *holdsAt* are soft-constrained. Both $MLN-EC_{SI^h}$ and $MLN-EC'_{SI^h}$ cases share the same knowledge base. In the $MLN-EC_{SI^h}$ case, the occurrence of SDEs causes none of the inertia clauses to become trivially satisfied. In the $MLN-EC'_{SI^h}$ case, however, the SDEs are randomly generated and cause a different subset of inertia clauses to become trivially satisfied at each time-point. In both cases the probability of the CE is reduced. In contrast to $MLN-EC_{SI^h}$, however, the inertia in $MLN-EC'_{SI^h}$ drops more sharply, as some of the clauses in Σ' are trivially satisfied by the given SDEs. Additionally, the probability of the CE to hold in $MLN-EC'_{SI^h}$ persists at a different level in each time-point, since different subsets of clauses become trivially satisfied each time. Similarly, in Figure 4(b) the rules about the inertia of $\neg \text{holdsAt}$ are soft-constrained. In contrast to $MLN-EC_{SI^{-h}}$, the occurrence of SDEs leads to the partial satisfaction of the initiation conditions causing the inertia in $MLN-EC'_{SI^{-h}}$ to persist with a different confidence at each time-point, increasing the probability of the CE to hold more sharply.

Having analysed the effect of softening the inertia rules, it is worth noting that in many real cases the entire knowledge base may be soft-constrained. In this case, since the rules in Σ are soft-constrained, CEs are not being initiated or terminated with absolute certainty. At the same time, CEs do not persist with certainty, as the rules in Σ' are also soft-constrained.

Depending on the requirements of the target application, various policies regarding the soft-constraining of the knowledge base may be adopted. This flexibility is one of the advantages of combining logic with probabilities in the proposed method. Furthermore, it should be stressed that in a typical event recognition application the knowledge base will contain a large number of clauses. The strength of a constraint imposed by a clause is also affected by the weights of other clauses with which it shares the same predicates. Due to these interdependencies, the manual setting of weights is bound to be suboptimal and cumbersome. Fortunately, the weights can be estimated automatically from training sets, using standard parameter optimisation methods.

5. EVALUATION

In this section we evaluate the proposed method ($MLN-EC$) in the domain of video activity recognition. We use the publicly available benchmark dataset of the CAVIAR project. The aim of the experiments is to assess the effectiveness of $MLN-EC$ in recog-

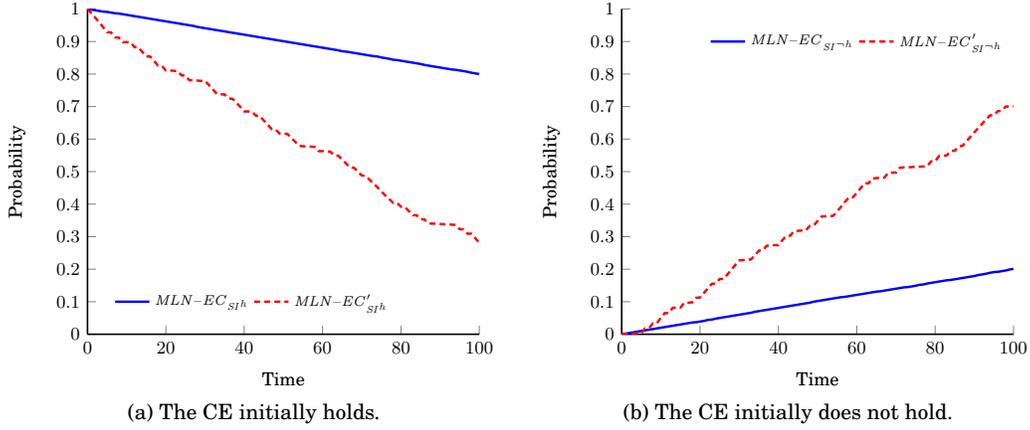


Fig. 4: In both figures no useful SDEs occur in the interval 0 to 100. In both $MLN-EC_{SI^h}$ and $MLN-EC'_{SI^h}$ none of the inertia clauses in Σ' become trivially satisfied by the SDEs. However, in $MLN-EC_{SI^h}$ and $MLN-EC'_{SI^h}$ some inertia clauses are trivially satisfied by the SDE. In the left figure the CE holds at time 0 with absolute certainty, while in the right figure the CE does not hold at time 0.

nising CEs that occur among people, based on imperfect CE definitions and in the presence of incomplete narratives of SDEs.

$MLN-EC$ combines the benefits of logic-based representation (e.g. direct expression of domain background knowledge) with probabilistic modelling (e.g. uncertainty handling). For comparison purposes, we include in the experiments two approaches that are closely related to our method. First, we include the logic-based activity recognition method of Artikis et al. [2010b], which we call here EC_{crisp} . Like our method, EC_{crisp} employs a variant of the Event Calculus and uses the same definitions of CEs. Unlike $MLN-EC$, EC_{crisp} cannot perform probabilistic reasoning. Second, we include a pure probabilistic method that employs a linear-chain Conditional Random Field model [Lafferty et al. 2001], which we call here $l-CRF$. Similar to our method, $l-CRF$ is a log-linear model that performs probabilistic reasoning over an undirected probabilistic network. On the other hand, $l-CRF$ does not employ a logic-based representation of the domain knowledge.

5.1. Setup

From the 28 videos of the CAVIAR dataset, we have extracted 19 sequences that are annotated with the *meeting* and/or *moving* CEs. The rest of the sequences in the dataset are ignored, as they do not contain examples of the two target CEs. Out of 19 sequences, 8 are annotated with both *moving* and *meeting* activities, 9 are annotated only with *moving* and 2 only with *meeting*. The total length of the extracted sequences is 12869 frames. Each frame is annotated with the occurrence or absence of a CE and is considered an example instance. The whole dataset contains a total of 25738 annotated example instances. There are 6272 example instances in which *moving* occurs and 3622 in which *meeting* occurs. Consequently, for both CEs the number of negative examples is significantly larger than the number of positive examples, 19466 for moving and 22116 for meeting.

The input of all three methods consists of a sequence of SDEs, i.e. *active*, *inactive*, *walking*, *running*, *enter* and *exit*. In both $MLN-EC$ and $l-CRF$, the spatial constraints *close* and *orientationMove* are precomputed and their truth value is provided as input.

In situations where no event occurs or the distance of the involved persons is above the highest predefined threshold, the tags *none* and *far* are given to l -CRF, respectively.

The output of the EC_{crisp} method consists of a sequence of ground *holdsAt* predicates, indicating which CEs are recognised. Since EC_{crisp} performs crisp reasoning, all CEs are recognised with absolute certainty. On the other hand, the output of the probabilistic methods depends on the inference type, i.e. maximum a-posteriori (MAP) or marginal inference. Given a sequence of SDEs, MAP inference outputs the most probable instantiations of CEs for all time-points. On the other hand, marginal inference estimates the probability of each CEs for all time-points.

Table II presents the structure of the training sequences for the probabilistic methods. In particular, Table II(a) shows an example training sequence for $MLN-EC$. Each sequence is composed of input SDEs (ground *happens*), precomputed spatial constraints between pairs of people (ground *close* and *orientationMove*), as well as the corresponding CE annotations (ground *holdsAt*). Negated predicates in the training sequence state that the truth value of the corresponding predicate is *False*. Table II(b) shows the equivalent input training data for l -CRF. The random variables *Person A* and *Person B* represent the events that the two persons may perform at each time-point and variables *Close* and *Orientation Move* represent the spatial constraints between the two persons. Similar to $MLN-EC$, l -CRF does not contain any hidden variables between input SDEs and output CEs. As a result, training is fully supervised for both methods.

To estimate the weights in l -CRF, we use the quasi-Newton optimisation algorithm L-BFGS [Byrd et al. 1994]. For MAP and marginal inference we use the *Viterbi* and *Forward-Backward* algorithms [Culotta and McCallum 2004; Sutton and McCallum 2007]. In $MLN-EC$ we use the second-order Diagonal Newton method of Lowd and Domingos [2007] and perform marginal inference with the *MC-SAT* algorithm [Poon and Domingos 2006], taking 1000 samples for each sequence. We additionally perform max-margin training and MAP inference, using the method of Huynh and Mooney [2009]. In the experiments we use the open-source software packages *ALCHEMY* [2005] and *CRF++*⁴.

$MLN-EC$ is tested under three different scenarios ($MLN-EC_{HI}$, $MLN-EC_{SI^h}$ and $MLN-EC_{SI}$, see Table III for a description). In all three variants of $MLN-EC$, the rules in Σ are soft-constrained while the inertia rules in Σ' are either soft or hard.

Throughout the experimental analysis, the results for marginal inference are presented in terms of F_1 score for threshold values ranging between 0.0 and 1.0. Any CE with probability above the threshold is considered recognised. A snapshot of the performance using the threshold value 0.5, is presented in terms of true positives (TP), false positives (FP), false negatives (FN), precision, recall and F_1 score. Additionally, the overall performance for marginal inference is measured in terms of area under precision-recall curve (AUPRC). The number of true negatives in our experiments is significantly larger than the number of true positives. Similar to F_1 score, precision and recall, the AUPRC is insensitive to the number of true negatives. The evaluation results using MAP inference are presented in terms of true positives (TP), false positives (FP), false negatives (FN), precision, recall and F_1 . We perform 10-fold cross-validation and all reported experiment statistics are micro-averaged over the instances of recognised CEs in the 10 folds.

5.2. Network Size and Inference Times

The CE definitions of *moving* (rules (8)–(13)) and *meeting* (rules (3)–(7)) together with the domain-independent axioms of $MLN-EC$ (rules (14)–(17)) are preprocessed using

⁴<http://code.google.com/p/crfpp>

Table II: Example training sets for CE *moving*.

Simple Derived Events	Composite Events
...	...
<i>happens(walking(id₁), 100)</i>	
<i>happens(walking(id₂), 100)</i>	<i>holdsAt(moving(id₁, id₂), 100)</i>
<i>orientationMove(id₁, id₂, 100)</i>	<i>holdsAt(moving(id₂, id₁), 100)</i>
<i>close(id₁, id₂, 24, 100)</i>	
...	...
<i>happens(active(id₁), 101)</i>	
<i>happens(walking(id₂), 101)</i>	<i>holdsAt(moving(id₁, id₂), 101)</i>
<i>orientationMove(id₁, id₂, 101)</i>	<i>holdsAt(moving(id₂, id₁), 101)</i>
\neg <i>close(id₁, id₂, 24, 101)</i>	
...	...
<i>happens(walking(id₁), 200)</i>	
<i>happens(running(id₂), 200)</i>	\neg <i>holdsAt(moving(id₁, id₂), 200)</i>
\neg <i>orientationMove(id₁, id₂, 200)</i>	\neg <i>holdsAt(moving(id₂, id₁), 200)</i>
\neg <i>close(id₁, id₂, 24, 200)</i>	
...	...

(a) Input narrative for *MLN-EC*.

Person A	Person B	Close	Orientation Move	Composite Events
...
walking	walking	24	True	Moving
active	walking	Far	True	NotMoving
...
walking	running	Far	False	NotMoving
...

(b) Input sequence for *l-CRF*.

Table II(a) shows a training set for *MLN-EC*. The first column is composed of a narrative of SDEs and precomputed spatial constraints for *MLN-EC*, while the second column contains the CE annotation in the form of ground *holdsAt* predicates. Table II(b) shows the equivalent training set for *l-CRF*. The first four columns contain the input SDEs and spatial constraints, while the last column contains the annotation.

Table III: Variants of *MLN-EC*, using hard and soft inertia rules in Σ' .

Scenarios	Description
<i>MLN-EC_{HI}</i>	All inertia rules in Σ' are hard-constrained.
<i>MLN-EC_{SI^h}</i>	The inertia rules of <i>holdsAt</i> are soft-constrained, while the rest of Σ' remains hard-constrained.
<i>MLN-EC_{SI}</i>	All inertia rules in Σ' are soft-constrained.

the transformation procedure presented in Section 3.3. Given the SDEs of all 19 sequences, the knowledge bases of *moving* and *meeting* are transformed into Markov networks of 128689 ground clauses and 51548 ground predicates.

Our experiments were performed on a machine with an Intel® Core™ i7-980 3.33 GHz processor and 24GB of RAM, running Debian GNU/Linux 7.0. Table IV shows the inference times of all *MLN-EC* scenarios: we show the inference times for the 12869 video frames of all 19 sequences (this corresponds to a video of 8 minutes and 58 seconds), as well as, the average inference time and standard deviation for a single frame.

Table IV: Probabilistic inference times of $MLN-EC$

Scenarios	Marginal		MAP	
	All sequences	Single frame	All sequences	Single frame
$MLN-EC_{HI}$	1 m 16 s	6.67 ± 2.42 ms	1 m 38 s	7.11 ± 1.47 ms
$MLN-EC_{SI^h}$	1 m 17 s	6.77 ± 2.32 ms	1 m 41 s	7.32 ± 1.40 ms
$MLN-EC_{SI}$	1 m 40 s	7.96 ± 3.07 ms	1 m 42 s	7.44 ± 1.47 ms

(a) *moving*

Scenarios	Marginal		MAP	
	All sequences	Single frame	All sequences	Single frame
$MLN-EC_{HI}$	1 m 21 s	6.32 ± 1.43 ms	0 m 19 s	1.40 ± 0.29 ms
$MLN-EC_{SI^h}$	1 m 15 s	6.06 ± 1.17 ms	0 m 19 s	1.41 ± 0.28 ms
$MLN-EC_{SI}$	1 m 30 s	7.16 ± 1.59 ms	0 m 20 s	1.48 ± 0.22 ms

(b) *meeting*

Our knowledge base transformation procedure reduces significantly the size and complexity of the produced ground Markov networks. Without this transformation procedure, the resulting Markov networks for the same 19 SDE sequences consist of 334452 ground clauses and 154428 ground predicates. In the case of *meeting* and the $MLN-EC_{HI}$ scenario, the time for marginal inference is 4 minutes and 33 seconds — this is more than 3 times slower than the inference over the transformed knowledge base (see $MLN-EC_{HI}$ in Table IV(b)).

5.3. The Methods Being Compared

Both EC_{crisp} and $MLN-EC$ employ a logic-based representation, implement a variant of the Event Calculus and contain equivalent definitions of CEs. The CE definitions of *meeting* and *moving* of EC_{crisp} are translated into first-order logic for $MLN-EC$ using the formulation proposed in Section 3.2. The definition of *meeting* is given by formulas (3)–(7), while that of *moving* is given by formulas (8)–(13). In contrast to EC_{crisp} , each clause in $MLN-EC$ may be associated with a weight value, indicating a degree of confidence.

Similar to $MLN-EC$, $l-CRF$ is a discriminative probabilistic graphical model. The relationship among CEs at successive time-points is modelled as a Markov network, conditioned on the input evidence of SDEs. A CE at any time-point in the sequence is represented by a Boolean random variable, stating whether the CE holds or not. For example, the random variables representing the *moving* CE may take either the tag value *Moving* or *NotMoving* at some time-point in the sequence.

However, there are also several differences between the two probabilistic methods. In $l-CRF$, the input SDEs and the spatial constraints are represented by multivariate random variables. For instance, the input SDEs for a particular person are represented by a single random variable that can take any SDE tag value, e.g. *active*, *inactive*, *walking*, etc. The relationship among random variables is defined by two types of features. The former type associates input SDEs and spatial constraints with output CEs at the same time-point, creating features for all possible instantiations. The latter type associates successive CEs, in order to form linear chains. In particular, features are constructed for each possible pair of CE instantiations at successive time-points. All features in $l-CRF$ are associated with weights and thus all relationships are soft-constrained.

On the other hand, $MLN-EC$ employs a logic-based representation and all features are produced from ground clauses. Domain knowledge is combined with the Event

Calculus axioms, in order to form the structure of the network. For example, the relations between successive CE instantiations are formed by the inertia axioms and the corresponding initiation and termination rules. Moreover, the *MLN-EC* provides control over the behaviour of CE persistence, by allowing the inertia clauses to be defined either as soft or as hard constraints. The probabilistic inference in *MLN-EC* can deal with both deterministic (i.e. hard-constrained clauses) and probabilistic (i.e. soft-constrained clauses) dependencies, as well as arbitrary structured networks. On the other hand, the structural simplicity of *l-CRF* allows for specialised and significantly faster inference and learning methods.

5.4. Experimental Results of the Crisp Event Calculus

The CE definitions are domain-dependent rules that together with the domain-independent axioms of the Event Calculus represent common sense domain knowledge. This knowledge may deviate from that implied by an annotated dataset, resulting in errors when recognising events. Such issues can be clearly shown by analysing the performance of *EC_{crisp}*, which uses the CE definitions also used in *MLN-EC* and does not involve the representation of probabilistic knowledge.

As shown in Figure 5, *EC_{crisp}* achieves a similar F_1 score for both activities. Figure 5 displays the results of marginal inference for the probabilistic techniques. However, in terms of precision and recall the situation is quite different, revealing two different cases of imperfect CE definitions. The precision for *moving* is 22 percentage points higher than that of *meeting*. The opposite holds for recall, with the recall for *moving* being 21.6 percentage points lower than that of *meeting*. The lower recall values for *moving* indicate a larger number of unrecognised *moving* activities (FN). In some example sequences *moving* is being initiated late, producing many false negatives. Additionally, the termination rules of *moving* cause the CE to be prematurely terminated in some cases. For example, when the distance of two persons that are moving together becomes greater than 34 pixels for a few frames, rule (9) terminates *moving*. On the other hand, compared to *moving*, the definition of *meeting* results in a larger number of erroneously recognised *meeting* activities (FP). The initiation rule (4), for example, causes the *meeting* activity to be initiated earlier than it should.

Another issue caused by the definitions of *meeting* and *moving* is that the two CEs may overlap. According to rules (3)–(13), the initiation of *moving* does not cause the termination of *meeting*. Consider, for example, a situation where two people meet for a while and thereafter they move together. During the interval in which *moving* is detected, *meeting* will also remain detected, as it is not terminated and the law of inertia holds. However, according to the annotation of the CAVIAR team these activities do not happen concurrently. Furthermore, *meeting* appears to be annotated in cases where the relative distance of both interacting persons is greater than that used in the initiation rules (3) and (4).

On the other hand, the domain knowledge may describe situations that are not included in the dataset. For example, by allowing the *meeting* CE to be initiated when the two persons are not very close to each other, one may achieve better results in this subset of the dataset, but erroneously recognise the *meeting* CE in other situations, e.g. people passing by each other. Additionally, the domain-independent property of inertia, which is included in the background knowledge, helps the method to continue to recognise the occurrence of a CE even when the narrative of SDEs is temporally incomplete, e.g. due to camera failures.

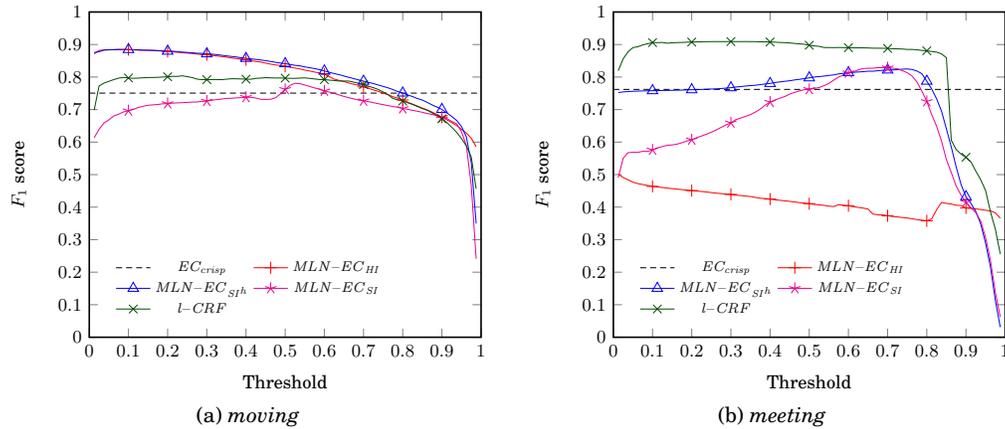


Fig. 5: F_1 scores using different threshold values for the *moving* and *meeting* CE.

5.5. Experimental Results of the Probabilistic Methods

The experiments for the probabilistic methods are organised into two tasks⁵. In the first task, both probabilistic methods are trained discriminatively and their performance is evaluated using 10-fold cross-validation. In the second task, we assess the value of inertia by erasing input evidence from randomly chosen successive time-points. We use the trained models from the first task, but the testing sequences are incomplete narratives of SDEs.

5.5.1. Task 1. In contrast to EC_{crisp} , $l-CRF$ is a probabilistic model and cannot employ background knowledge in the form of common sense rules. The recognition of a CE is affected by all input SDEs that are detected at a particular time-point, as well as from the adjacent CEs that have been recognised. The parameters of the model are estimated from the training set and thus the model is completely data driven. Compared to EC_{crisp} , $l-CRF$ achieves better performance for both *moving* and *meeting*. Using marginal inference, $l-CRF$ gives higher F_1 scores for most threshold values, as shown in Figures 5(a) and 5(b). For threshold 0.5, the F_1 score is higher than of EC_{crisp} by 4.6 and 13.5 percentage points for *moving* and *meeting*, respectively (see Tables V(a) and V(c)). The recall of *moving* is higher by 18.4 percentage points, while the precision is lower by 13.7 percentage points. $l-CRF$ recognises a larger number of *moving* activities, increasing the number of both true and false positives. As noted in Section 5.4, this can be achieved by looser spatial constraints. Recall and precision scores for *meeting* are higher by 1.7 and 23.7 percentage points, respectively. In the case of MAP inference, $l-CRF$ gives almost the same F_1 score as EC_{crisp} for *moving* (see Tables VI(a) and VI(b)). In general $l-CRF$ outperforms EC_{crisp} in both CE. Unlike EC_{crisp} , $l-CRF$ interrupts the recognition of *meeting* when *moving* starts.

In the first $MLN-EC$ scenario, indicated by $MLN-EC_{HI}$, rules in Σ are soft-constrained, i.e. they are associated with a weight value after training. Those weights control the certainty with which a CE holds when its initiation or termination conditions are satisfied. The rules in Σ' , however, remain hard-constrained and thus the behaviour of inertia for both CEs is preserved deterministically and cannot be adjusted. Compared to EC_{crisp} , $MLN-EC_{HI}$ achieves a higher F_1 score using marginal

⁵The MLN and $l-CRF$ formatted version of the dataset, the CE definitions of $MLN-EC$, the template files of $l-CRF$ and the results of both probabilistic methods can be found at: <http://www.iit.demokritos.gr/~anskarl/pub/mlnec>

Table V: Results for the *moving* and *meeting* CE using marginal inference.

Method	TP	TN	FP	FN	Precision	Recall	F ₁ score	AUPRC
<i>EC_{crisp}</i>	4008	19086	400	2264	0.9093	0.6390	0.7506	
<i>MLN-EC_{HI}</i>	5121	18584	902	1151	0.8502	0.8165	0.8330	0.8847
<i>MLN-EC_{SJ^h}</i>	5233	18542	944	1039	0.8472	0.8343	0.8407	0.8597
<i>MLN-EC_{SI}</i>	4938	17760	1726	1334	0.7410	0.7873	0.7635	0.8280
<i>l-CRF</i>	5160	17964	1522	1112	0.7722	0.8227	0.7967	0.8358

(a) *moving*, threshold 0.5.(b) *moving*

Method	TP	TN	FP	FN	Precision	Recall	F ₁ score	AUPRC
<i>EC_{crisp}</i>	3099	20723	1413	523	0.6868	0.8556	0.7620	
<i>MLN-EC_{HI}</i>	1284	20786	1350	2338	0.4875	0.3545	0.4105	0.5160
<i>MLN-EC_{SJ^h}</i>	3060	21147	989	562	0.7557	0.8448	0.7978	0.7559
<i>MLN-EC_{SI}</i>	3052	20800	1336	570	0.6955	0.8426	0.7620	0.7730
<i>l-CRF</i>	3160	21874	262	462	0.9234	0.8724	0.8972	0.8937

(c) *meeting*, threshold 0.5.(d) *meeting*Table VI: Results for the *moving* and *meeting* CE using MAP inference

Method	TP	TN	FP	FN	Precision	Recall	F ₁ score
<i>EC_{crisp}</i>	4008	19086	400	2264	0.9093	0.6390	0.7506
<i>MLN-EC_{HI}</i>	5598	18358	1128	674	0.8323	0.8925	0.8614
<i>MLN-EC_{SJ^h}</i>	5902	18398	1088	370	0.8443	0.9410	0.8901
<i>MLN-EC_{SI}</i>	4040	17911	1575	2232	0.7195	0.6441	0.6797
<i>l-CRF</i>	4716	17848	1638	1556	0.7422	0.7519	0.7470

(a) *moving*

Method	TP	TN	FP	FN	Precision	Recall	F ₁ score
<i>EC_{crisp}</i>	3099	20723	1413	523	0.6868	0.8556	0.7620
<i>MLN-EC_{HI}</i>	3099	20739	1397	523	0.6893	0.8556	0.7635
<i>MLN-EC_{SJ^h}</i>	3067	21825	311	555	0.9079	0.8468	0.8763
<i>MLN-EC_{SI}</i>	1083	21641	495	2539	0.6863	0.2990	0.4165
<i>l-CRF</i>	3154	21906	230	468	0.9320	0.8708	0.9004

(b) *meeting*

inference for the *moving* CE, for most threshold values (see Figure 5(a)). For threshold 0.5, the recall of *MLN-EC* is higher by 17.7 percentage points than *EC_{crisp}* while precision is lower by 6 points, leading the F_1 score of *MLN-EC* to be higher than *EC_{crisp}* by 8.2 points (Table V(a)). This improvement in recall is caused by the weights that are learned for the termination conditions, which prevent the *moving* CE from terminating prematurely. Compared to *l-CRF*, *MLN-EC_{HI}* achieves better F_1 scores for many thresholds and higher AUPRC by 4.8 percentage points (see Tables V(a) and V(b)). Using MAP inference, *MLN-EC_{HI}* achieves higher F_1 score by 11 percentage points than both *EC_{crisp}* and *l-CRF* (see Table VI(a)). Compared to *EC_{crisp}*, the recall of *MLN-EC_{HI}* is improved by 25.3 percentage points, while its precision drops by 7 percentage points. *MLN-EC_{HI}* achieves higher recall and precision scores than *l-CRF*, by 14 and 9 percentage points, respectively. However, in the case of *meeting*, *MLN-EC_{HI}* performs worse than *EC_{crisp}* and *l-CRF* in marginal inference, as shown in Figure 5(b) and Table V(c). The combination of hard-constrained inertia rules with the fact that *meeting* does not terminate when *moving* starts, push the weights of the initiation rules to very low values during training. This situation results in many unrecognised

meeting instances and low precision and recall values. Max-margin training in this scenario is not affected as much as the Diagonal Newton weight learning method, leading to a model with similar behaviour and performance as EC_{crisp} , as shown in Table VI(b).

In the $MLN-EC_{SI^h}$ scenario, while Σ remains soft-constrained, the inertia rules of *holdsAt* in Σ' are also soft-constrained. As a result, the probability of a CE tends to decrease, even when the required termination conditions are not met and nothing relevant is happening. This scenario is more suitable to our target activity recognition task and $MLN-EC$ learns a model with a high F_1 score for both CEs. In order to explain the effect of soft constraining the inertia of *holdsAt*, we will use again the example of *meeting* being recognised and thereafter *moving* being also recognised. Since *meeting* is not terminated, it continues to hold and overlaps with *moving*. During the overlap, all occurring SDEs are irrelevant with respect to *meeting* and cannot cause any initiation or termination. As a result, the recognition probability of *meeting* cannot be reinforced, by re-initiation. As shown in Section 4, in such circumstances the recognition probability of *meeting* gradually decreases.

For the *moving* CE, the performance of $MLN-EC_{SI^h}$ using marginal inference is similar to $MLN-EC_{HI}$, as shown in Figure 5(a) and Table V(a). Using a threshold of 0.5, recall is higher than that of EC_{crisp} by 19.5 percentage points while precision is lower by 6.2 points, resulting in 9 points increase in F_1 measure. Compared to $l-CRF$, $MLN-EC_{HI}$ achieves higher F_1 scores for many thresholds (see Figure 5(a)), as well as higher AUPRC by 2.4 percentage points (see Table V(b)). In the case of MAP inference, $MLN-EC_{SI^h}$ increases further its performance (Table VI(a)). Compared to EC_{crisp} , recall is higher by 30 percentage points and precision drops only by 6.5 percentage points, resulting in 14 percentage points higher F_1 score. $MLN-EC_{SI^h}$ achieves higher F_1 score, precision and recall than $l-CRF$ by 14.3, 10 and 18.9 percentage points, respectively.

For the *meeting* CE, the performance of $MLN-EC_{SI^h}$ using marginal inference is significantly better than of $MLN-EC_{HI}$ (see Figure 5(b)). At the 0.5 threshold value, precision increases by 6.9 percentage points over EC_{crisp} , while recall falls by only 1 point and thus F_1 is higher by 3.5 points (Table V(c)). However, the F_1 scores of $MLN-EC_{SI^h}$ remain lower than those of $l-CRF$ and its AUPRC is lower by 13.8 points (Table V(d)). Using MAP inference, the performance of $MLN-EC$ improves, but remains worse than $l-CRF$, by 2.4 percentage points in terms of F_1 (Table VI(b)). $MLN-EC_{SI^h}$ performs similarly to $l-CRF$ for *meeting*. $MLN-EC_{SI^h}$ misses the recognition of *meeting* at time-points where the persons involved are not sufficiently close to each other according to the initiation rules (3) and (4), i.e. they have a distance greater than 25 pixels.

Finally, in the $MLN-EC_{SI}$ scenario, the entire knowledge base is soft-constrained. The weights in Σ allow full control over the confidence that a CE holds when its initiation or termination conditions are met. Additionally, by soft-constraining the rules in Σ' , $MLN-EC_{SI}$ provides full probabilistic inertia. However, this flexibility comes at the cost of an increase in the number of parameters to be estimated from data, as all clauses in the knowledge base are now soft-constrained. As a result, $MLN-EC_{SI}$ requires more training data. Using marginal inference $MLN-EC_{SI}$ performs almost the same as EC_{crisp} in terms of F_1 score, but worse than $MLN-EC_{SI^h}$ for both CEs. In the case of MAP inference, $MLN-EC_{SI}$ performs worse than EC_{crisp} , as well as $l-CRF$ for both CEs.

The three variants of $MLN-EC$ used in the above experiments, illustrated the potential benefits of softening the constraints and performing probabilistic inference in event recognition. In contrast to EC_{crisp} , an important characteristic of $MLN-EC$ is that multiple successive initiations (or terminations) can increase (or decrease) the recognition probability of a CE. By softening the CE definitions, premature initiation or termination can be avoided. In particular, as explained above, the weights learned

for the termination definitions of the *moving* CE reduced the number of unrecognised *moving* activities.

The choice of rules to be softened affects significantly the event recognition accuracy. In the presented application, for example, the $MLN-EC_{SI^h}$ setting is the best choice, as softening the inertia of *holdsAt* provides advantages over crisp recognition. Depending on the target application and the availability of training data, different types of inertia rules may be softened, varying the inertia behaviour from deterministic to completely probabilistic. This is a key feature of $MLN-EC$.

5.5.2. Task II. An important difference between the proposed logic-based approach and its purely data-driven competitors, such as *l-CRF*, is that it is less dependent on the peculiarities of the training data. The use of background knowledge about the task and the domain, in terms of logic, can make the recognition process more robust to variations in the data. Such variations are very common in practice, particularly in dynamic environments, such as the ones encountered in event recognition. The common assumption made in machine learning that the training and test data share the same statistical properties is often violated in these situations. In order to measure the benefits that we can gain by the combination of background knowledge and learning in MLNs, we examine the recognition of CEs under such situations. In particular, in this second task, we are comparing the performance of $MLN-EC$ and *l-CRF* in cases where the input evidence is missing from a number of successive time-points in the test data. The models are not retrained and thus their weights remain as they were estimated in Task I.

The incomplete test sequences are generated artificially by erasing successive input SDEs and associated spatial relations at random starting points. We have generated two variants of incomplete test sequences, one containing random blank intervals of length 10 and 20 time-points respectively. The starting time-points of the blank intervals are chosen randomly with probability 0.01, drawn from a uniform distribution. Since the target CEs require the interaction of two persons, erasing events that involve just a single person cannot affect the performance of the recognition methods that we compare. Therefore, blank intervals are created only from time-points where both persons are involved in some SDEs. This process of artificially generating incomplete test sequences is repeated five times, generating corresponding test sets.

The recognition performance can be affected in various ways, depending on the position where evidence is erased from the test sequences. For example, when the beginning of an activity is erased, its recognition will be delayed, increasing the number of false negatives. Similarly, erasing information at the end of an activity will delay the termination of a CE, resulting in a higher number of false positives. In cases where missing information appears during an activity, the recognition of the CE may be interrupted, increasing the number of false negatives. The recognition performance may even be improved in some cases, due to the removal of SDEs that would cause false positives or false negatives.

Out of all the methods examined in Task I the performance of EC_{crisp} and $MLN-EC_{HI}$ is bound to be affected less by the missing data, due to the use of deterministic inertia. This is because the erased evidence will often be in the interval that a CE is (not) being recognised. In these cases, the erased evidence will not affect the inertia of the CE and the CE will remain (not) recognised. EC_{crisp} and $MLN-EC_{HI}$ are only affected when the evidence is erased at the beginning or at the end of an activity, which is less frequent. For this reason, we chose to exclude these two methods from Task II. Furthermore, we excluded $MLN-EC_{SI}$, which performed significantly worse than $MLN-EC_{SI^h}$ in Task I. Therefore, in this task we compare only $MLN-EC_{SI^h}$ against *l-CRF*.

In the rest of this section we will denote as *medium* and *large incomplete sequences* the test sequences that contain random blank intervals of 10 and 20 time-points respectively. The evaluation measures are the same as in Task I. Figure 6 presents the results in terms of F_1 score for the two methods, using marginal inference. The bar charts of Figure 7, on the other hand, present the average AUPRC of the two methods compared also to the AUPRC when no data is removed (from Tables V(b) and V(d) of Task I). The threshold values range between 0.0 and 1.0. Using a similar illustration, Figures 8, 9 and 10 present the results of the two methods using MAP inference. All results are averaged over five runs and error bars correspond to the standard deviation.

Unlike $MLN-EC_{SI^h}$, $l-CRF$ appears to be affected significantly from incomplete evidence. Using marginal inference on the *moving* CE in the original test sequences (see Table V(a)) $l-CRF$ achieved an F_1 score of 0.7967. At the same threshold value, the average F_1 score of $l-CRF$ drops to 0.566 and 0.53 for medium and large incomplete sequences, respectively. $MLN-EC_{SI^h}$ is affected much less, achieving F_1 scores of 0.836 and 0.832, for medium and large incomplete sequences, respectively. In terms of AUPRC (Figure 7(a)), the performance of $l-CRF$ also drops by 13 points, while $MLN-EC_{SI^h}$ is almost unaffected. When MAP inference is used, the effect of the removal of data seems to be larger. The recall of $l-CRF$ falls by more than 45 percentage points, causing the F_1 score to drop by more than 30 percentage points (Figures 9(a) and 8(a)). The number of recognised *moving* activities is reduced, resulting in an increase in precision, but with high variance (Figure 9(a)). The precision of $MLN-EC_{SI^h}$ remains close to the original test set, with a small variance. However, its recall drops and causes the reduction of F_1 score by 8 and 9 percentage points for medium and large incomplete sequences, respectively.

In the case of the *meeting* CE, $MLN-EC_{SI^h}$ also seems to resist more than $l-CRF$ to the distortion of the data. The F_1 score is higher than that of $l-CRF$ for many threshold values, using marginal inference (see Figures 6(b) and 6(d)). For threshold 0.5 in the original test sequences $l-CRF$ achieved an F_1 score that was higher by 10 percentage points than that of $MLN-EC_{SI^h}$. However, when data are removed its F_1 score for the same threshold drops much lower than that of $MLN-EC_{SI^h}$ (a difference of more than 15 percentage points). The AUPRC of $l-CRF$ also drops much more (more than 10 points) than that of $MLN-EC_{SI^h}$ (see Figure 7(b)). The effect is even higher when MAP inference is used for *meeting* CE. In particular, the recall of $l-CRF$ drops more than 60 percentage points (see Figure 10(b)), while that of $MLN-EC_{SI^h}$ drops much less. Thus, the F_1 score of $MLN-EC_{SI^h}$ reduces by less than 10 percentage points, while that of $l-CRF$ is 50 percentage points lower than in the original data (see Figure 8(b)).

In summary, this task confirmed that the logic-based $MLN-EC$ method is more robust than its purely statistical $l-CRF$ counterpart, when data are removed from the test set, rendering it less similar to the training set. This is due to the fact that $l-CRF$ is completely data-driven and does not employ explicit domain knowledge. On the other hand, $MLN-EC$ employs background knowledge, including the domain independent axioms of inertia. Consequently, the persistence of a CE is modelled differently by $l-CRF$ and $MLN-EC$. $l-CRF$ learns to maintain the state of CEs under some circumstances that appear in the training data. However it does not model the inertia explicitly. Therefore, when the circumstances change, its performance is hurt significantly. In contrast, $MLN-EC$ enjoys the explicit modelling of inertia, provided as background knowledge. Even when this inertia is softened, it remains a strong bias in the model. As a result, $MLN-EC$ avoids overfitting the training data and behaves robustly when the data changes.

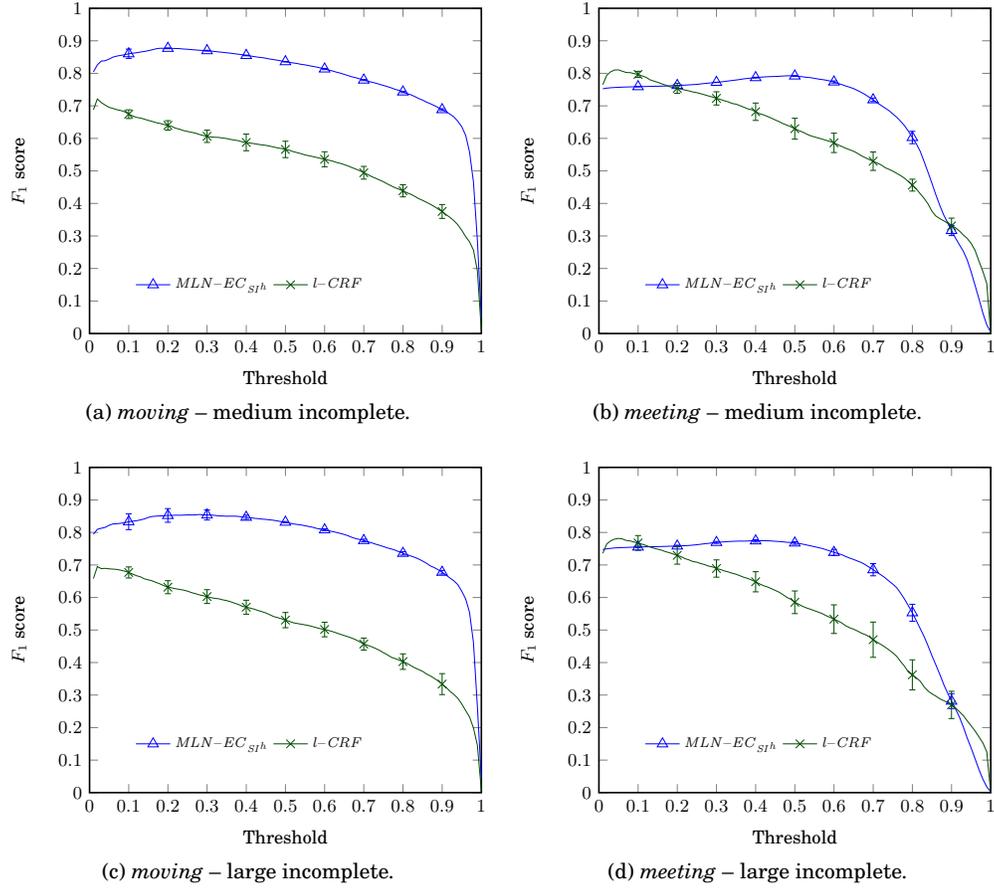


Fig. 6: Average F_1 scores over five runs when data are removed. Marginal inference is used.

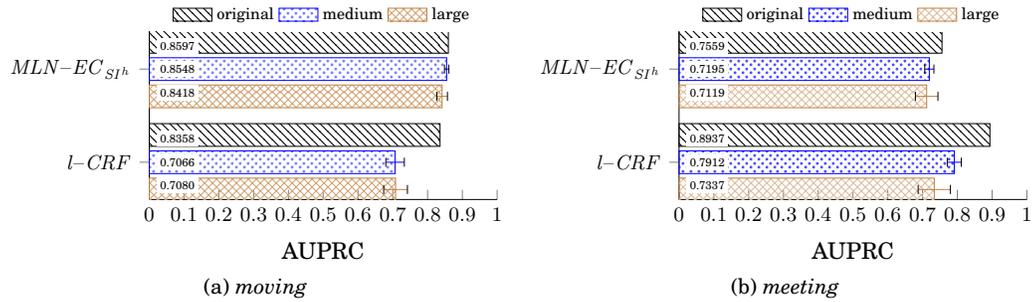


Fig. 7: Average AUPRC scores over five runs when data are removed. Marginal inference is used.

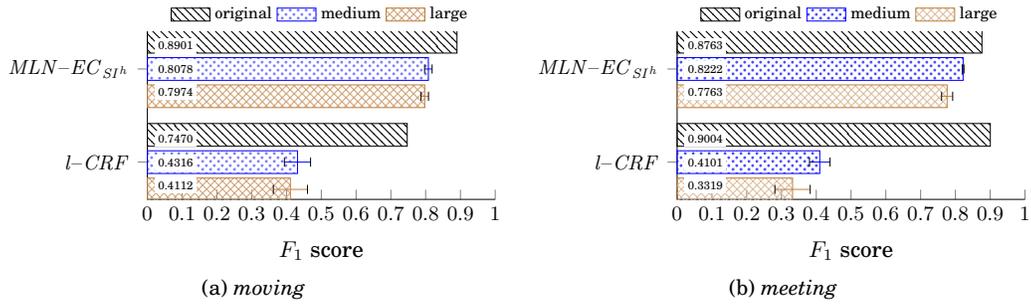


Fig. 8: Average F_1 scores over five runs when data are removed. MAP inference is used.

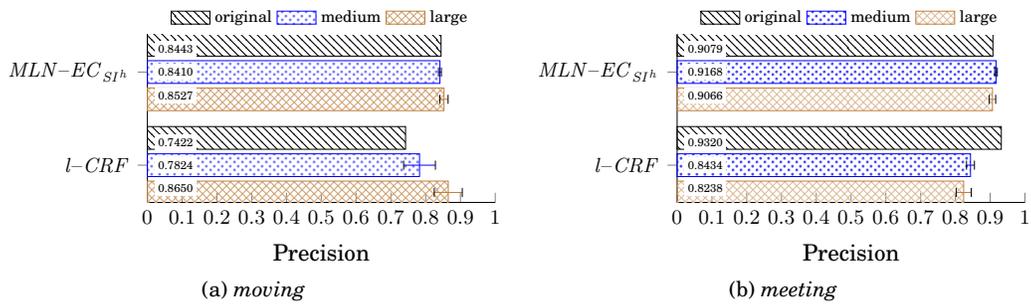


Fig. 9: Average precision over five runs when data are removed. MAP inference is used.

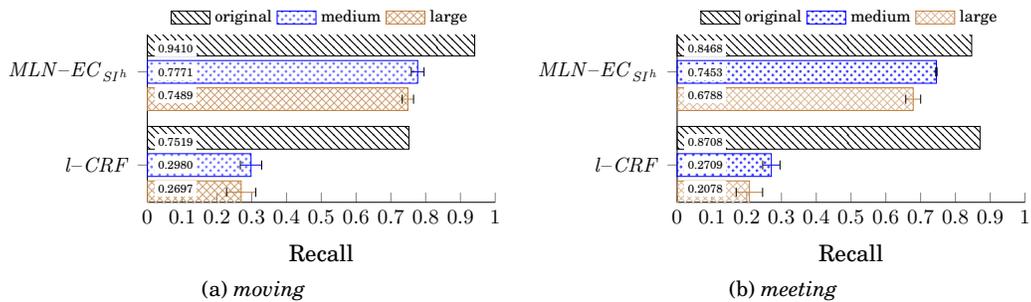


Fig. 10: Average recall over five runs when data are removed. MAP inference is used.

6. RELATED WORK

Event Calculus is related to other formalisms in the literature of common sense reasoning, such as the Situation Calculus [McCarthy and Hayes 1968; Reiter 2001], the Fluent Calculus [Thielscher 1999; 2001], the action language $C+$ [Giunchiglia et al. 2004; Akman et al. 2004] and Temporal Action Logics [Doherty et al. 1998; Kvarnström 2005]. Action formalisms provide domain-independent axioms in order to represent and reason about the effects of events and support the property of inertia. Comparisons and proofs of equivalence between formalisms for commonsense reasoning can be found in Kowalski and Sadri [1997], Van Belleghem et al. [1997], Chittaro and

Montanari [2000], Miller and Shanahan [2002], Schiffel and Thielscher [2006], Mueller [2006, Chapter 15], Craven [2006] and Paschke and Kozlenkov [2009].

Probabilistic extensions of the Situation Calculus have been proposed in the literature, in order to support noisy input from sensors, stochastic events and model Markov Decision Processes, e.g. see Bacchus et al. [1995], Pinto et al. [2000], Mateus et al. [2001], Hajishirzi and Amir [2008] and Reiter [2001, Chapter 12]. Furthermore, Hölldobler et al. [2006] proposed a probabilistic extension of the Fluent Calculus. Both the Situation Calculus and the Fluent Calculus, as well as in their probabilistic variants use a tree model of time, in which each event may give rise to a different possible future. A point in time is represented by a situation, which is a possible sequence of events. As a result, events are represented to occur sequentially and atemporally.

In the Event Calculus, as well as in $\mathcal{C}+$ and Temporal Action Logics, there is a single time-line on which events occur. This is a more suitable model for event recognition, where the task is to recognise CEs of interest in a time-stamped sequence of observed SDEs.

$\mathcal{PC}+$ is a probabilistic generalisation of $\mathcal{C}+$ that incorporates probabilistic knowledge about the effects of events [Eiter and Lukasiewicz 2003]. $\mathcal{PC}+$ supports non-deterministic and probabilistic effects of events, as well as probabilistic uncertainty about the initial state of the application. Similar to the aforementioned probabilistic variants of common sense reasoning languages, the method focuses on planning under uncertainty while inertia remains deterministic.

In $MLN-EC$ we can have customisable inertia behaviour by adjusting the weights of the inertia axioms, as shown in Sections 4 and 5. To deal with uncertainty we employ the framework of Markov Logic Networks for probabilistic modelling and automatically estimate the weights from training data.

A related approach that we have developed in parallel is that of Skarlatidis et al. [2013]. The method employs an Event Calculus formalism that is based on probabilistic logic programming [Kimmig et al. 2011] and handles noise in the input data. Input SDEs are assumed to be independent and are associated with detection probabilities. In contrast, the MLN approach presented in this paper does not make any independence assumption about the input SDEs. The Event Calculus axioms and CE definitions in the knowledge base remain hard-constrained. Given a narrative of SDEs, a CE may be recognised with some probability. Any initiation or termination caused by the given SDEs increases or decreases the probability of a CE to hold. Inertia is modelled by the closed-world semantics of logic programming and is restricted to be deterministic.

Albanese et al. [2010] proposed a probabilistic activity description language for expressing CE on top of an image processing suit. Their method merges logic-based formulation with probabilistic modelling. The input SDEs can be either deterministic or probabilistic. CE are defined by users in a first-order logic where the dependencies between SDEs are modelled by triangular norms [Fagin 1996]. The method provides efficient algorithms for off-line and on-line recognition. Another logic-based method for handling uncertainty in activity recognition is proposed by Shet et al. [2007]. The method employs logic programming and handles uncertainty using the Bilattice framework [Ginsberg 1988]. The knowledge base consists of domain-specific rules, expressing CEs in terms of SDEs. Each CE or SDE is associated with two uncertainty values, indicating a degree of information and confidence respectively. The underlying idea of the method is that the more confident information is provided, the stronger the belief about the corresponding CE becomes. Filippaki et al. [2011] proposed a logic-based method that recognises user activities over noisy or incomplete data. The method recognises CEs from SDEs using rules that impose temporal and spatial constraints between SDEs. Some of the constraints in CE definitions are optional. As a result, a

CE can be recognised from incomplete information, but with lower confidence. The confidence of a CE increases when more of the optional SDEs are recognised. Due to noisy or incomplete information, the recognised CEs may be logically inconsistent with each other. The method resolves those inconsistencies using the confidence, duration and number of involved SDEs. In contrast to these methods, our work employs MLNs that have formal probabilistic semantics and an Event Calculus formalism. Our method can represent complex CEs and take advantage of the domain-independent property of inertia.

The Chronicle Recognition System [Dousson and Maigat 2007], is a symbolic event recognition method that has been applied to a variety of problems, such as cardiac arrhythmia recognition [Callens et al. 2008], computer network monitoring [Dousson 1996; Dousson and Maigat 2007] and airport ground traffic monitoring [Choppy et al. 2009]. CEs are represented using a declarative temporal language and translated into temporal constraint networks, in order to perform efficient event recognition. A temporal constraint network is a directed symbolic network, where vertices correspond to instantiations of some event (CE or SDE) and edges represent temporal constraints between the involved events. Similarly, in the domain of video interpretation, the scenario recognition method [Vu et al. 2003] translates CE definitions into temporal constraint networks. For reasons of efficiency the networks are automatically decomposed into several sub-networks. The Chronicle Recognition System language has also been translated into Petri-Nets which are executed to recognise CEs [Choppy et al. 2009]. These symbolic methods, however, cannot handle uncertainty which naturally exists in many real-world applications and may seriously compromise event recognition accuracy.

Albanese et al. [2007] developed a recognition method that models activities using a stochastic automaton language. In [Albanese et al. 2011b], the representation is extended with temporal constraints, where each subsequent SDE can occur within a user-defined temporal interval. Using possible-worlds based modelling, the method finds totally (or partially) unexplained activities, i.e. instances of activities for which no model exists yet. A probabilistic extension to Petri-Nets has been proposed by Albanese et al. [2008]. The method can represent complex interactions with multiple persons, temporal constraints and handle input uncertainty. Our probabilistic method employs a first-order logic representation and the Event Calculus, in order to represent complex CE. As presented in Section 5, our method can handle the uncertainty in the CE definitions and perform robustly when input SDEs are incomplete. Based on the generic framework of MLN, our method can benefit from a variety of probabilistic inference and Machine Learning algorithms.

Probabilistic graphical models have been successfully applied to a variety of event recognition tasks where a significant amount of uncertainty exists. Since event recognition requires the processing of streams of time-stamped SDEs, numerous event recognition methods are based on sequential variants of probabilistic graphical models, such as Hidden Markov Models (HMM) [Rabiner and Juang 1986], Dynamic Bayesian Networks (DBN) [Murphy 2002] and linear-chain Conditional Random Fields (CRF) [Lafferty et al. 2001]. Such models can naturally handle uncertainty but their propositional structure provides limited representation capabilities. To overcome this limitation, graphical models have been extended to model interactions between multiple entities [Brand et al. 1997; Gong and Xiang 2003; Wu et al. 2007; Vail et al. 2007], capture long-term dependencies between states [Hongeng and Nevatia 2003] and model the hierarchical composition of events [Natarajan and Nevatia 2007; Liao et al. 2005]. However, the lack of a formal representation language makes the definition of structured CEs complicated and the use of background knowledge very hard.

Recently, statistical relational learning (SRL) methods have been applied to event recognition. These methods combine logic with probabilistic models, in order to represent complex relational structures and perform reasoning under uncertainty. Using a declarative language as a template, SRL methods specify probabilistic models at an abstract level. Given an input stream of SDE observations, the template is partially or completely instantiated, creating lifted or propositional graphical models on which probabilistic inference is performed [de Salvo Braz et al. 2008; Raedt and Kersting 2010].

Among others, HMMs have been extended in order to represent states and transitions using logical expressions [Kersting et al. 2006; Natarajan et al. 2008]. In contrast to standard HMM, the logical representation allows the model to represent compactly probability distributions over sequences of logical atoms, rather than propositional symbols. Similarly, DBNs have been extended using first-order logic [Manfredotti 2009; Manfredotti et al. 2010]. A tree structure is used, where each node corresponds to a first-order logic expression, e.g. a predicate representing a CE, and can be related to nodes of the same or previous time instances. Compared to their propositional counterparts, the extended HMM and DBN methods can compactly represent CE that involve various entities.

Our method is based on Markov Logic Networks (MLNs), which is a more general and expressive model. The knowledge base of weighted first-order logic formulas in MLNs defines an arbitrarily structured undirected graphical model. Therefore, MLNs provide a generic SRL framework, which subsumes various graphical models, e.g. HMM, CRF, etc., and can be used with expressive logic-based formalisms, such as the Event Calculus. The inertia axioms of our method allow the model to capture long-term dependencies between events. Additionally, adopting a discriminative model, the method avoids common independence assumptions over the input SDEs.

Markov Logic Networks have recently been used for event recognition in the literature. Biswas et al. [2007] combine the information provided by different low-level classifiers with the use of MLNs, in order to recognise CEs. Tran and Davis [2008]; Kembhavi et al. [2010] take into account the confidence value of the input SDEs, which may be due to noisy sensors. A more expressive approach that can represent persistent and concurrent CEs, as well as their starting and ending points, is proposed by Helaoui et al. [2011]. However, that method has a quadratic complexity to the number of time-points.

Morariu and Davis [2011] proposed an MLN-based method that uses interval relations. The method determines the most consistent sequence of CEs, based on the observations of low-level classifiers. Similar to Tran and Davis [2008]; Kembhavi et al. [2010] the method expresses CEs in first-order logic, but it also employs temporal relations from the Interval Algebra [Allen 1983]. In order to avoid the combinatorial explosion of possible intervals, as well as to eliminate the existential quantifiers in CE definitions, a bottom-up process eliminates the unlikely CE hypotheses. The elimination process can only be applied to domain-dependent axioms, as it is guided by the observations and the Interval Algebra relations. A different approach to interval-based activity recognition is the Probabilistic Event Logic [Brendel et al. 2011; Selman et al. 2011]. Similar to MLNs, the method defines a log-linear model from a set of weighted formulas, but the formulas are represented in Event Logic [Siskind 2001]. Each formula defines a soft constraint over some events, using interval relations that are represented by the *spanning intervals* data structure. The method performs inference via a local-search algorithm (based on MaxWalkSAT of Kautz et al. [1997]) and avoids grounding all possible time intervals by using the spanning intervals structure. In our work, we address the combinatorial explosion problem in a more generic manner, through the efficient representation of the domain-independent axioms. Ad-

ditionally, we use a transformation procedure to further simplify the structure of the Markov network. The transformation is performed at the level of the knowledge base and is independent of the input SDEs.

Sadilek and Kautz [2012] employ hybrid-MLNs [Wang and Domingos 2008] in order to recognise successful and failed interactions between humans, using noisy location data from GPS devices. The method uses hybrid formulas that denoise the location data. Hybrid formulas are defined as normal soft-constrained formulas, but their weights are also associated with a real-valued function, e.g. the distance of two persons. As a result, the strength of the constraint that a hybrid rule imposes is defined by both its weight and function — e.g. the closer the distance, the stronger the constraint. The weights are estimated from training data. The method does not employ any generic formalism for representing the events and their effects and thus it uses only domain-dependent CE definitions. On the other hand, the use of a hybrid approach for numeric constraints is an interesting alternative to the discretisation adopted by our method.

7. CONCLUSIONS

We addressed the issue of imperfect CE definitions that stems from the uncertainty that naturally exists in event recognition. The proposed solution is a probabilistic version of the Event Calculus based on Markov Logic Networks (*MLN-EC*). The method has declarative and formal (probabilistic) semantics, inheriting the properties of the Event Calculus. We placed particular emphasis on the efficiency and effectiveness of our approach. By simplifying the axioms of the Event Calculus, as well as following a knowledge transformation procedure, the method produces compact Markov networks of reduced complexity. Consequently, the performance of probabilistic inference is improved, as it takes place on a simpler model. *MLN-EC* supports flexible CE persistence, ranging from deterministic to probabilistic, in order to meet the requirements of different applications. Due to the use of MLNs, the method lends itself naturally to learning the weights of event definitions from data, as the manual setting of weights is sub-optimal and cumbersome. *MLN-EC* is trained discriminatively, using a supervised learning technique. In the experimental evaluation, *MLN-EC* outperforms its crisp equivalent on a benchmark data. *MLN-EC* matches the performance of a linear-chain Conditional Random Fields method. Furthermore, due to the use of background knowledge in the form of Event Calculus, *MLN-EC* is affected less by missing data in the test sequences than its probabilistic akin.

There are several directions in which we would like to extend our work. In many applications the input SDE observations are accompanied by a degree of confidence, usually in the form of probability. Therefore, we consider extending our method in order to exploit data that involves such confidence values, either in the form of additional clauses (e.g. Tran and Davis [2008], Morariu and Davis [2011]), or by employing different inference algorithms (e.g. Jain and Beetz [2010]). Furthermore, we would like to address the problems that involve numerical constraints by adopting a hybrid-MLNs (e.g. Sadilek and Kautz [2012]) or a similar approach. We also consider extending our formalism in order to support temporal interval relations, using preprocessing techniques (e.g. Morariu and Davis [2011]), or by employing different representation and inference methods (e.g. Brendel et al. [2011], Selman et al. [2011]). As shown in Section 5, the *MLN-EC* with soft-constrained inertia performs well. We would like to extend our method to automatically soften the right subset of inertia axioms. In Section 5.5.2, the *MLN-EC* was shown to perform well on using incomplete SDEs. We would like to extend this line of work and propose a method to estimate the missing SDEs, possibly through probabilistic abduction (e.g. Singla and Mooney [2011], Blythe et al. [2011]). We also plan to evaluate our method and its extensions using additional data from

different domains, e.g. cyber security [Albanese et al. 2011a]. Finally, we would like to examine structure learning/refinement methods for the CE definitions, since they are often hard to acquire from experts.

ACKNOWLEDGMENTS

The authors wish to thank Professor Larry S. Davis and Dr. Vlad I. Morariu for their valuable comments and suggestions, as well as, the EC Funded CAVIAR project/IST 2001 37540 for the publicly available activity recognition dataset.

REFERENCES

- Varol Akman, Joohyung Lee Selim T. Erdoğan, Vladimir Lifschitz, and Hudson Turner. 2004. Representing the Zoo World and the Traffic World in the language of the Causal Calculator. *Artificial Intelligence* 153, 1 (2004), 105–140.
- Massimiliano Albanese, Rama Chellappa, Naresh Cuntoor, Vincenzo Moscato, Antonio Picariello, V. S. Subrahmanian, and Octavian Udrea. 2010. PADS: A Probabilistic Activity Detection Framework for Video Data. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 12 (2010), 2246–2261. DOI : <http://dx.doi.org/10.1109/TPAMI.2010.33>
- Massimiliano Albanese, Rama Chellappa, Naresh P. Cuntoor, Vincenzo Moscato, Antonio Picariello, V. S. Subrahmanian, and Octavian Udrea. 2008. A Constrained Probabilistic Petri Net Framework for Human Activity Detection in Video. *IEEE Transactions on Multimedia* 10, 6 (2008), 982–996.
- Massimiliano Albanese, Sushil Jajodia, Andrea Pugliese, and V. S. Subrahmanian. 2011a. Scalable Detection of Cyber Attacks. In *In Proceedings of the 10th International Conference of Computer Information Systems - Analysis and Technologies*. 9–18.
- Massimiliano Albanese, Cristian Molinaro, Fabio Persia, Antonio Picariello, and V. S. Subrahmanian. 2011b. Finding “Unexplained” Activities in Video. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*. 1628–1634.
- Massimiliano Albanese, Vincenzo Moscato, Antonio Picariello, V. S. Subrahmanian, and Octavian Udrea. 2007. Detecting Stochastically Scheduled Activities in Video. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*. 1802–1807.
- James F. Allen. 1983. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM* 26, 11 (1983), 832–843.
- Alexander Artikis, Marek Sergot, and Georgios Paliouras. 2010a. A Logic Programming Approach to Activity Recognition. In *Proceedings of the 2nd International Workshop on Events in Multimedia (EiMM)*. ACM, 3–8.
- Alexander Artikis, Anastasios Skarlatidis, and Georgios Paliouras. 2010b. Behaviour Recognition from Video Content: a Logic Programming Approach. *International Journal on Artificial Intelligence Tools (JAIT)* 19, 2 (2010), 193–209.
- Alexander Artikis, Anastasios Skarlatidis, François Portet, and Georgios Paliouras. 2012. Logic-based event recognition. *Knowledge Engineering Review* 27, 4 (2012), 469–506.
- Fahiem Bacchus, Joseph Y. Halpern, and Hector J. Levesque. 1995. Reasoning about Noisy Sensors in the Situation Calculus. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*. Morgan Kaufmann, 1933–1940.
- Marenglen Biba, Fatos Xhafa, Floriana Esposito, and Stefano Ferilli. 2011. Engineering SLS Algorithms for Statistical Relational Models. In *Proceedings of the International Conference on Complex, Intelligent and Software Intensive Systems (CISIS)*. IEEE Computer Society, 502–507.
- Rahul Biswas, Sebastian Thrun, and Kikuo Fujimura. 2007. Recognizing Activities with Multiple Cues. In *Proceedings of the 2nd Workshop on Human Motion - Understanding, Modeling, Capture and Animation (Lecture Notes in Computer Science)*. Springer, 255–270.
- Hendrik Blockeel. 2011. Statistical Relational Learning. In *Handbook on Neural Information Processing*, Monica Bianchini, Marco Maggini, and Lakhmi Jain (Eds.). Springer.
- James Blythe, Jerry R. Hobbs, Pedro Domingos, Rohit J. Kate, and Raymond J. Mooney. 2011. Implementing Weighted Abduction in Markov Logic. In *In Proceedings of the International Conference on Computational Semantics*. Oxford, England, 55–64.
- Matthew Brand, Nuria Oliver, and Alex Pentland. 1997. Coupled hidden Markov models for complex action recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 994–999.

- William Brendel, Alan Fern, and Sinisa Todorovic. 2011. Probabilistic Event Logic for Interval-based Event Recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 3329–3336.
- Richard H Byrd, Jorge Nocedal, and Robert B Schnabel. 1994. Representations of quasi-Newton matrices and their use in limited memory methods. *Mathematical Programming* 63, 1 (1994), 129–156.
- Lucie Callens, Guy Carrault, Marie-Odile Cordier, Elisa Fromont, François Portet, and René Quiniou. 2008. Intelligent adaptive monitoring for cardiac surveillance. In *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI) (Frontiers in Artificial Intelligence and Applications)*, Vol. 178. IOS Press, 653–657.
- Luca Chittaro and Angelo Montanari. 2000. Temporal Representation and Reasoning in Artificial Intelligence: Issues and Approaches. *Annals of Mathematics and Artificial Intelligence* 28, 1 (2000), 47–106.
- Christine Choppy, Olivier Bertrand, and Patrice Carle. 2009. Coloured Petri Nets for Chronicle Recognition. In *Proceedings of the 14th International Conference on Reliable Software Technologies - Ada-Europe (Lecture Notes in Computer Science)*, Vol. 5570. Springer, 266–281.
- Michael Collins. 2002. Discriminative training methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In *Proceedings of the ACL Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Vol. 10. Association for Computational Linguistics, 1–8.
- Robert Craven. 2006. *Execution mechanisms for the action language C+*. Ph.D. Dissertation. University of London.
- Aron Culotta and Andrew McCallum. 2004. Confidence estimation for information extraction. In *Proceedings of HLT-NAACL 2004: Short Papers*. Association for Computational Linguistics, 109–112.
- Rodrigo de Salvo Braz, Eyal Amir, and Dan Roth. 2008. A Survey of First-Order Probabilistic Models. In *Innovations in Bayesian Networks: Theory and Applications*. Studies in Computational Intelligence, Vol. 156. Springer, 289–317.
- Patrick Doherty, Joakim Gustafsson, Lars Karlsson, and Jonas Kvarnström. 1998. TAL: Temporal Action Logics Language Specification and Tutorial. *Electronic Transactions on Artificial Intelligence* 2, 3–4 (1998), 273–306.
- Patrick Doherty, Witold Lukaszewicz, and Andrzej Szalas. 1997. Computing Circumscription Revisited: A Reduction Algorithm. *Journal of Automated Reasoning* 18, 3 (1997), 297–336.
- Pedro Domingos and Daniel Lowd. 2009. *Markov Logic: An Interface Layer for Artificial Intelligence*. Morgan & Claypool Publishers.
- Christophe Dousson. 1996. Alarm Driven Supervision for Telecommunication Network: II-On-line Chronicle Recognition. *Annals of Telecommunications* 51, 9 (1996), 501–508.
- Christophe Dousson and Pierre Le Maigat. 2007. Chronicle Recognition Improvement Using Temporal Focusing and Hierarchization. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*. 324–329.
- Thomas Eiter and Thomas Lukasiewicz. 2003. Probabilistic Reasoning about Actions in Nonmonotonic Causal Theories. In *UAI*, Christopher Meek and Uffe Kjærulff (Eds.). Morgan Kaufmann, 192–199.
- Opher Etzion and Peter Niblett. 2010. *Event Processing in Action*. Manning Publications Company. I–XXIV, 1–360 pages.
- Ronald Fagin. 1996. Combining Fuzzy Information from Multiple Systems. In *In Proceedings of the Fifteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. ACM Press, 216–226.
- Chrysi Filippaki, Grigoris Antoniou, and Ioannis Tsamardinos. 2011. Using Constraint Optimization for Conflict Resolution and Detail Control in Activity Recognition. In *Proceedings of the 2nd International Joint Conference on Ambient Intelligence (AmI) (Lecture Notes in Computer Science)*, Vol. 7040. Springer, 51–60.
- Avigdor Gal, Segev Wasserkrug, and Opher Etzion. 2011. Event Processing over Uncertain Data. In *Reasoning in Event-Based Distributed Systems*, Sven Helmer, Alexandra Poulouvasilis, and Fatos Xhafa (Eds.). Studies in Computational Intelligence, Vol. 347. Springer, 279–304.
- Matthew L. Ginsberg. 1988. Multivalued logics: a uniform approach to reasoning in artificial intelligence. *Computational Intelligence* 4 (1988), 265–316.
- Enrico Giunchiglia, Joohyung Lee, Vladimir Lifschitz, Norman McCain, and Hudson Turner. 2004. Non-monotonic Causal Theories. *Artificial Intelligence* 153, 1 (2004), 49–104.
- Shaogang Gong and Tao Xiang. 2003. Recognition of Group Activities using Dynamic Probabilistic Networks. In *Proceedings of the 9th International Conference on Computer Vision (ICCV)*, Vol. 2. IEEE Computer Society, 742–749.
- Hannaneh Hajishirzi and Eyal Amir. 2008. Sampling First Order Logical Particles. In *Proceedings of the 24th Conference in Uncertainty in Artificial Intelligence (UAI), Helsinki, Finland*. AUAI Press, 248–255.

- Rim Helaoui, Mathias Niepert, and Heiner Stuckenschmidt. 2011. Recognizing Interleaved and Concurrent Activities: A Statistical-Relational Approach. In *Proceedings of the 9th Annual International Conference on Pervasive Computing and Communications (PerCom)*. IEEE Computer Society, 1–9.
- Steffen Hölldobler, Eldar Karabaev, and Olga Skvortsova. 2006. FLUCAP: a heuristic search planner for first-order MDPs. *Journal of Artificial Intelligence Research (JAIR)* 27, 1 (Dec. 2006), 419–439.
- Somboon Hongeng and Ramakant Nevatia. 2003. Large-Scale Event Detection Using Semi-Hidden Markov Models. In *Proceedings of the 9th International Conference on Computer Vision (ICCV)*, Vol. 2. IEEE Computer Society, 1455–1462.
- Tuyen N. Huynh and Raymond J. Mooney. 2009. Max-Margin Weight Learning for Markov Logic Networks. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD) (Lecture Notes in Computer Science)*, Vol. 5781. Springer, 564–579.
- Tuyen N. Huynh and Raymond J. Mooney. 2011. Online Max-Margin Weight Learning for Markov Logic Networks. In *Proceedings of the 11th SIAM International Conference on Data Mining (SDM11)*. Mesa, Arizona, USA, 642–651.
- Dominik Jain and Michael Beetz. 2010. Soft Evidential Update via Markov Chain Monte Carlo Inference. In *Proceedings of the 33rd Annual German Conference on AI (KI) (Lecture Notes in Computer Science)*, Vol. 6359. Springer, 280–290.
- Henry Kautz, Bart Selman, and Yueyen Jiang. 1997. A General Stochastic Approach to Solving Problems with Hard and Soft Constraints. In *The Satisfiability Problem: Theory and Applications*, Dingzhu Gu, Jun Du, and Panos Pardalos (Eds.). DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 35. AMS, 573–586.
- Aniruddha Kembhavi, Tom Yeh, and Larry S. Davis. 2010. Why Did the Person Cross the Road (There)? Scene Understanding Using Probabilistic Logic Models and Common Sense Reasoning. In *Proceedings of the 11th European Conference on Computer Vision (ECCV) (Lecture Notes in Computer Science)*, Vol. 6312. Springer, 693–706.
- Kristian Kersting, Babak Ahmadi, and Sriraam Natarajan. 2009. Counting Belief Propagation. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI)*. AUAI Press, 277–284.
- Kristian Kersting, Luc De Raedt, and Tapani Raiko. 2006. Logical Hidden Markov Models. *Journal of Artificial Intelligence Research (JAIR)* 25, 1 (2006), 425–456.
- A. Kimmig, B. Demoen, L. De Raedt, V. Santos Costa, and R. Rocha. 2011. On the Implementation of the Probabilistic Logic Programming Language ProbLog. *Theory and Practice of Logic Programming* 11 (2011), 235–262.
- Stanley Kok, Parag Singla, Matthew Richardson, Pedro Domingos, Marc Sumner, Hoifung Poon, and Daniel Lowd. 2005. *The Alchemy system for statistical relational AI*. Technical Report. Department of Computer Science and Engineering, University of Washington, Seattle, WA. <http://alchemy.cs.washington.edu>.
- Robert Kowalski and Fariba Sadri. 1997. Reconciling the Event Calculus with the Situation Calculus. *The Journal of Logic Programming* 31, 1 (1997), 39–58.
- Robert Kowalski and Marek Sergot. 1986. A Logic-based Calculus of Events. *New Generation Computing* 4, 1 (1986), 67–95.
- Jonas Kvarnström. 2005. *TALplanner and Other Extensions to Temporal Action Logic*. Ph.D. Dissertation. Linköping.
- John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the 18th International Conference on Machine Learning (ICML)*. Morgan Kaufmann, 282–289.
- Lin Liao, Dieter Fox, and Henry A. Kautz. 2005. Hierarchical Conditional Random Fields for GPS-Based Activity Recognition. In *International Symposium of Robotics Research (ISRR) (Springer Tracts in Advanced Robotics (STAR))*, Vol. 28. Springer, 487–506.
- Vladimir Lifschitz. 1994. Circumscription. In *Handbook of logic in Artificial Intelligence and Logic Programming*. Vol. 3. Oxford University Press, Inc., 297–352.
- Thor List, José Bins, Jose Vazquez, and Robert B. Fisher. 2005. Performance Evaluating the Evaluator. In *2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*. 129–136. DOI : <http://dx.doi.org/10.1109/VSPETS.2005.1570907>
- Daniel Lowd and Pedro Domingos. 2007. Efficient Weight Learning for Markov Logic Networks. In *Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD) (Lecture Notes in Computer Science)*, Vol. 4702. Springer, 200–211.
- David C. Luckham. 2002. *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley Longman Publishing Co., Inc.

- David C. Luckham. 2011. *Event Processing for Business: Organizing the Real-Time Enterprise*. Wiley.
- Cristina Manfredotti. 2009. Modeling and Inference with Relational Dynamic Bayesian Networks. In *Advances in Artificial Intelligence*, Yong Gao and Nathalie Japkowicz (Eds.). Lecture Notes in Computer Science, Vol. 5549. Springer Berlin / Heidelberg, 287–290.
- Cristina Manfredotti, Howard Hamilton, and Sandra Zilles. 2010. Learning RDBNs for Activity Recognition. In *NIPS Workshop on Learning and Planning from Batch Time Series Data*.
- Paulo Mateus, António Pacheco, Javier Pinto, Amílcar Sernadas, and Cristina Sernadas. 2001. Probabilistic Situation Calculus. *Annals of Mathematics and Artificial Intelligence* 32, 1 (2001), 393–431.
- John McCarthy. 1980. Circumscription - A Form of Non-Monotonic Reasoning. *Artificial Intelligence* 13, 1-2 (1980), 27–39.
- John McCarthy and Patrik J. Hayes. 1968. *Some philosophical problems from the standpoint of artificial intelligence*. Stanford University.
- Rob Miller and Murray Shanahan. 2002. Some Alternative Formulations of the Event Calculus. In *Computational Logic: Logic Programming and Beyond, Essays in Honour of Robert A. Kowalski, Part II (Lecture Notes in Computer Science)*. Springer, 452–490.
- Tom Minka. 2005. *Discriminative models, not discriminative training*. Technical Report. Microsoft Research. Available at: <http://research.microsoft.com/pubs/70229/tr-2005-144.pdf>.
- Vlad I. Morariu and Larry S. Davis. 2011. Multi-agent event recognition in structured scenarios. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 3289–3296.
- Erik T. Mueller. 2006. *Commonsense Reasoning*. Morgan Kaufmann.
- Erik T. Mueller. 2008. Event Calculus. In *Handbook of Knowledge Representation. Foundations of Artificial Intelligence*, Vol. 3. Elsevier, 671–708.
- Kevin P. Murphy. 2002. *Dynamic Bayesian Networks: representation, inference and learning*. Ph.D. Dissertation. University of California.
- Pradeep Natarajan and Ramakant Nevatia. 2007. Hierarchical Multi-channel Hidden Semi Markov Models. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*. 2562–2567.
- Sriram Natarajan, Hung Hai Bui, Prasad Tadepalli, Kristian Kersting, and Weng-Keen Wong. 2008. Logical Hierarchical Hidden Markov Models for Modeling User Activities. In *Proceedings of the 18th International Conference Inductive Logic Programming (ILP) (Lecture Notes in Computer Science)*, Vol. 5194. Springer, 192–209.
- Adrian Paschke and Alexander Kozlenkov. 2009. Rule-Based Event Processing and Reaction Rules. In *Proceedings of the 3rd International Symposium on Rules (RuleML) (Lecture Notes in Computer Science)*, Vol. 5858. Springer, 53–66.
- J. Pinto, A. Sernadas, C. Sernadas, and P. Mateus. 2000. Non-determinism and uncertainty in the Situation Calculus. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 8, 2 (2000), 127–149.
- Hoifung Poon and Pedro Domingos. 2006. Sound and Efficient Inference with Probabilistic and Deterministic Dependencies. In *Proceedings of the 21st AAAI Conference on Artificial Intelligence*. AAAI Press, 458–463.
- Lawrence R. Rabiner and Biing-Hwang Juang. 1986. An introduction to Hidden Markov Models. *Acoustics, Speech, and Signal Processing Magazine (ASSP)* 3, 1 (1986), 4–16.
- Luc De Raedt and Kristian Kersting. 2010. Statistical Relational Learning. In *Encyclopedia of Machine Learning*, Claude Sammut and Geoffrey I. Webb (Eds.). Springer, 916–924.
- Raymond Reiter. 2001. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. MIT Press.
- Sebastian Riedel. 2008. Improving the Accuracy and Efficiency of MAP Inference for Markov Logic. In *Proceedings of the 24th Conference in Uncertainty in Artificial Intelligence (UAI)*. AUAI Press, 468–475.
- Adam Sadilek and Henry A. Kautz. 2012. Location-Based Reasoning about Complex Multi-Agent Behavior. *Journal of Artificial Intelligence Research (JAIR)* 43 (2012), 87–133.
- Stephan Schiffel and Michael Thielscher. 2006. Reconciling Situation Calculus and Fluent Calculus. In *Proceedings of the National Conference on Artificial Intelligence*, Vol. 21. AAAI Press, 287.
- Joseph Selman, Mohamed R. Amer, Alan Fern, and Sinisa Todorovic. 2011. PEL-CNF: Probabilistic event logic conjunctive normal form for video interpretation. In *Proceedings of the International Conference on Computer Vision Workshops (ICCVW)*. IEEE Computer Society, 680–687.
- Murray Shanahan. 1997. *Solving the Frame Problem: A Mathematical Investigation of the Common Sense Law of Inertia*. MIT Press.

- Murray Shanahan. 1999. The Event Calculus Explained. In *Artificial Intelligence Today*, Michael Wooldridge and Manuela Veloso (Eds.). Lecture Notes in Computer Science, Vol. 1600. Springer, 409–430.
- Vinay D. Shet, Jan Neumann, Visvanathan Ramesh, and Larry S. Davis. 2007. Bilattice-based Logical Reasoning for Human Detection. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 1–8.
- Parag Singla and Pedro Domingos. 2005. Discriminative Training of Markov Logic Networks. In *Proceedings of the 20th National Conference on Artificial Intelligence*. AAAI Press / The MIT Press, 868–873.
- Parag Singla and Pedro Domingos. 2006. Memory-Efficient Inference in Relational Domains. In *Proceedings of the 21st AAAI Conference on Artificial Intelligence*. AAAI Press, 488–493.
- Parag Singla and Pedro Domingos. 2008. Lifted First-Order Belief Propagation. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*. AAAI Press, 1094–1099.
- Parag Singla and Raymond J. Mooney. 2011. Abductive Markov Logic for Plan Recognition. In *In Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*. AAAI Press, 1069–1075.
- Jeffrey Mark Siskind. 2001. Grounding the Lexical Semantics of Verbs in Visual Perception using Force Dynamics and Event Logic. *Journal of Artificial Intelligence Research (JAIR)* 15 (2001), 31–90.
- Anastasios Skarlatidis, Alexander Artikis, Jason Filippou, and Georgios Paliouras. 2013. A Probabilistic Logic Programming Event Calculus. *Journal of Theory and Practice of Logic Programming (TPLP)* (2013).
- Anastaios Skarlatidis, Georgios Paliouras, George Vouros, and Alexander Artikis. 2011. Probabilistic Event Calculus Based on Markov Logic Networks. In *Proceedings of the 5th International Symposium on Rules (RuleML) (Lecture Notes in Computer Science)*, Vol. 7018. Springer, 155–170.
- Charles Sutton and Andrew McCallum. 2007. An Introduction to Conditional Random Fields for Relational Learning. In *Introduction to Statistical Relational Learning*, Lise Getoor and Ben Taskar (Eds.). MIT Press, 93–127.
- Michael Thielscher. 1999. From Situation Calculus to Fluent Calculus: State update axioms as a solution to the inferential frame problem. *Artificial intelligence* 111, 1 (1999), 277–299.
- Michael Thielscher. 2001. The qualification problem: A solution to the problem of anomalous models. *Artificial Intelligence* 131, 1 (2001), 1–37.
- Son Dinh Tran and Larry S. Davis. 2008. Event Modeling and Recognition Using Markov Logic Networks. In *Proceedings of the 10th European Conference on Computer Vision (ECCV) (Lecture Notes in Computer Science)*, Vol. 5303. Springer, 610–623.
- Pavan Turaga, Rama. Chellappa, V. S. Subrahmanian, and Octavian Udrea. 2008. Machine Recognition of Human Activities: A Survey. *IEEE Transactions on Circuits and Systems for Video Technology*. 18, 11 (2008), 1473–1488.
- Douglas L. Vail, Manuela M. Veloso, and John D. Lafferty. 2007. Conditional Random Fields for Activity Recognition. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS)*. IFAAMAS, 1331–1338.
- K. Van Belleghem, M. Denecker, and D. De Schreye. 1997. On the relation between Situation Calculus and Event Calculus. *The Journal of Logic Programming* 31, 1 (1997), 3–37.
- Van-Think Vu, François Brémond, and Monique Thonnat. 2003. Automatic Video Interpretation: A Novel Algorithm for Temporal Scenario Recognition. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*. 1295–1302.
- Jue Wang and Pedro Domingos. 2008. Hybrid Markov Logic Networks. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*. AAAI Press, 1106–1111.
- Tsu-yu Wu, Chia-chun Lian, and Jane Yung-jen Hsu. 2007. Joint Recognition of Multiple Concurrent Activities using Factorial Conditional Random Fields. In *Proceedings of the Workshop on Plan, Activity, and Intent Recognition (PAIR)*. AAAI Press, 82–88.