Characterising Driver Intention via Hierarchical Perception-Action Modelling

***Abstract*—We seek a mechanism for the classification of the intentional behaviour of a cognitive agent, specifically a driver, in terms of a psychological Perception-Action (P-A) model, such that the resulting system would be potentially suitable for use in intelligent driver assistance. P-A models of human intentionality assume that a cognitive agent's perceptual domain is learned in response to the outcome of the agent's actions rather than vice versa. In this way, the perceptual domain is maintained at an appropriate level of complexity in relation to the agent's embodied motor capabilities, greatly simplifying visual processing. A *hierarchical* Perception-Action model further captures the subsumptive task-based hierarchicality implicit within human actions by assuming a parallel subsumption occur within the perceptual domain. Assuming this model enables us to characterise intentions at each level of the P-A hierarchy using a range of perceptual descriptors derived from the UK Highway Code and their correlation with driver gaze behaviour. The problem of classification is thus reconciling high-level protocols (i.e., Highway Code rules) with low-level features. We evaluate generative and discriminative logic-based methods for carrying out this classification based on the control, signal and motor inputs of an instrumented vehicle and find that generative model gives superior intentional classification performance due to the strongly-protocol driven nature of the driving environment.**

***Index Terms*—Perception-Action Modelling, Subsumption Architectures, Hierarchical systems, Human factors, Cognition.**

I. Introduction

THERE is an increasing recognition that the use of approaches based on human behaviour and inference of cognitive processes will be required for practical driver assistance systems since real environments are complex in nature, and human drivers play a major role in almost three quarters of all traffic accidents [1]. The objective of this paper is thus to propose a methodology based on a *Perception-Action* (P-A) model to characterise human (driver) intentions in a manner appropriate to the design of a cognitive driver assistance system and implement the proposed methodology using three distinct rule-based classification algorithms. We define our notion of *percepts* and *actions* as well as other key terms as follows:

Percepts are internal descriptors of observable objects (e.g., traffic signs, lights, pedestrians, are moving objects, lane boundaries etc.,) for an embodied cognitive agent.

Actions cause changes in *percepts* (e.g., eye-gaze, braking, acceleration, steering, signalling etc).

Intention is a planned action (anticipatory or compensatory) that is to be performed by the embodied agent (e.g., turning left, stopping, junction approach etc.,).

We use the principle of ***Bijectivity*** in order to link *percepts* to *actions* such that an *action* brings about a unique transition from one *percept* to another.

Being embodied in an environment, it is possible to consider human cognition in terms of the relationship between *actions* and *perceptions* [2]. In classical approaches to modelling intelligent robot behaviour (e.g., in CAD applications [3]), a fixed representational domain is assumed; however, a perception-action (P-A) framework for cognition [4] implicitly assumes that a *bijectivity* exists between actions and perceptual-transitions. In the P-A model there is hence an attempt to represent the world in the most efficient manner with respect to the ability of the cognitive system to bring about changes in it (amounting to an *affordance*-based modelling of the environment [5]). This bijectivity is with respect to all *potential* actions, so novel exploratory actions can be proposed and tested against the predicted perceptual outcome.

Novel percepts can thus be created and appended in a bottom-up manner when novel actions are proposed that subsume a set of existing actions (in the manner of [6]), so that a corresponding hierarchy of percepts is generated (see [7] for a simulated implementation of this approach). Intentional behaviour is thus typically characterised by a particular high-level perceptual goal that requires a series of sub tasks to be carried out, each with their own lower-level perceptual goals. Higher-level (i.e., more abstract) perceptions and actions are thus *grounded* (in the sense of Harnad [8]) through the hierarchy, with high-level actions implicitly generating appropriate contextualisation (motor orientation sub-tasks etc) at the lower levels, so that conceptualisation of actions at the higher level involves an autonomous scheduling of perception-action sub-tasks throughout the hierarchy [9]. In the following, we model a particular realisation of the P-A hierarchy assumed to operate in humans, termed the Extended Control Model (*ECOM*), which derives from psychological research [10] in the context of driving. The aim of this paper, is thus to find an appropriate mechanism for identifying the various levels of activity of these intentional task hierarchies by recognising the relevant percepts and actions employed by drivers in negotiating typical junction scenarios. (Junction scenarios are chosen so as to give full scope for this task-subsumption to become evident; evaluation is carried out with respect to expert annotation). This requires a car equipped with both eye-tracking and a forward camera, as well as the ability to log control and signal inputs, in order to obtain an indication of the relevant intentional behaviour.

The problem is thus one of (human driver) intention classification with respect to captured data, labelled via expert annotation as regards the intentional hierarchy. Since, at the higher levels, percepts are highly decontextualised, a hierarchical perception-action system is also a *symbol grounding* system [8], such that the higher levels are concerned with abstract symbol manipulation. (A key feature of P-A framework is that

there is no *a priori* requirement for global consistency in scene description at the lowest-level, only consistency through the hierarchy). Classification can therefore be approached from both a stochastic and deductive perspective, or a mixture of the two. This paper will set out to quantify which of these approaches is most appropriate to modelling human P-A hierarchies in the context of driving, with the potential for application to the problem of building cognitive driver assistance systems.

II. Conceptual Background

A. Hierarchical Perception-Action Models of Cognition

Gibson described *embodied* agents in terms of *affordances*, i.e., those possibilities offered by objects in the environment which are related to the agent's motor capabilities [5]. An (*embodied*) cognitive system is therefore expected to expand its knowledge by acquiring and storing information autonomously, effectively adapting and improving by reinforcing its understanding of the environment (e.g., by utilising a feedback loop), that might comprise active motor experimentation and (generally vision-based) assessment of the results. It should also have a certain degree of malleability and resilience towards unexpected events [11].

The classical approach towards modelling of the vision system in both artificial and human cognitive agents has tended to emphasize the representation of the scene (i.e., its operating environment) often in explicitly geometric terms prior to calculating actions. The described scene is then typically used, in artificial cognitive systems, for assignment of the scene objectives and the planning of actions. This model, although successful in certain circumstances, however fails to adapt to novel situations well. Granlund [4] points out that the classical approach of scene abstraction prior to action implementation results in a loss of important contextual qualifiers of a spatial and temporal nature, on the one hand, and an unnecessary degree of perceptual redundancy given the likely scene transitions on the other hand. He argues that Perception-Action systems constitute a much more robust approach, i.e., it supports active interpretation of the environment by relating *perceptual* changes to those *actions* that brought these changes (i.e., the principle of *bijectivity*). Thus a cognitive (vision) system utilising the perception action approach, instead of describing the scene in terms of physical parameters such as geometry of objects or scenes, rather builds up models of structures relating the percept domain to the agent's actions; we may say that 'action *precedes* perception' [4]. In Granlund's model, the higher level P-A modules implicitly derive symbolic representations from the lower-level modules by applying an appropriate degree of abstraction and decontextualisation.

Granlund's formulation of P-A system comprises P-A mappings at various hierarchical levels of scene abstraction in modelling of human cognitive behaviour (i.e., to classify of human intentions). Hierarchical modelling techniques have long been implicated in designing artificial cognitive systems [12] in order to address problems with classical control system theory (comprising a series of functional units) when applied

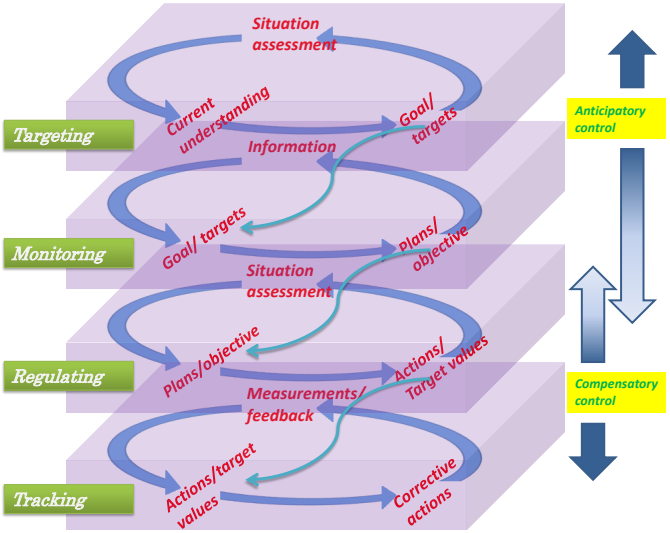

Fig. 1: The Extended Control Model (ECOM) [10].

to artificial cognitive systems. Brooks [6] described a more robust and flexible robot control system build using subsumptive task based hierarchical layers of asynchronous modules. Here, each layer or module corresponds to a simple computational unit, with higher layers subsuming the roles of lower layers by inhibiting their corresponding outputs, collectively forming a *subsumption hierarchy*. The current paper will assume the existence of a subsumptive hierarchy model in determining the distinct layers of the perception-action network present in human driving intentions, as such, we use a *hierarchical perception-action cognitive model* (i.e., **Extended Control Model**), for modelling of human intentions.

B. The Extended Control Model (ECOM)

Within the context of our problem, the assumed approach for behavioural modelling of human intentions with respect to the world is in terms of a subsumptive task-based hierarchical perception-action circuit, (thereby limiting the number of non-objectively correlated assumptions made regarding the nature of human cognition). Following psychological research, the Extended Control Model (ECOM) has been developed [10], which describes human cognitive performance in terms of four distinct (but simultaneous) layers of control (Fig. 1), three of which are appropriate to the current investigation: *Monitoring, Regulating & Tracking*. Note that we don't use the *Targeting* level, given that the experimental dataset does not comprise any route planning or navigational instructions to the driver of the car. Being a hierarchical perception-action methodology, ECOM assumes that the saliency of perceptual states depends upon the agents *actions* and uses this as the basis for modelling the environment. The three levels of ECOM and their relevance to current work are succinctly described as follows.

1) Monitoring level behaviour (being the highest of the three relevant levels of the ECOM hierarchy) in the car-driving case tends to keep a track of all the traffic signs and signals as well as road vehicle orientation and positions; it also sets objectives and activates plans for actions, such as monitoring the condition of the vehicle. Note that at the higher levels

of the ECOM hierarchy, actions are *abstract* and *protocol-governed* (i.e., based on Highway Code rules).

2) Regulating level provide input into the Tracking control-loop in order to perform specic, protocol relevant actions, (e.g., changing lane, positioning of a car relative to other objects on the road, or avoiding obstacles). The Regulating control level directs the tracking control level to perform a specific, high-way code relevant action, (e.g., changing lane). Other regulating intentions were limited to the following; intentionally stopping and turning right/left at a junction. It is also established that, where necessary, these regulating intentions would be linked hierarchically.

3) Tracking level behaviour is the lowest level in the ECOM hierarchy and describes the immediate responses of an agent to external perceptions in order to maintain the current state. It effectively manages the continuous activity undertaken to keep the vehicle within a specific, discrete conceptual configuration or logical state (e.g., car-order within a lane). From a drivers perspective this manifests itself as minor modifications of car speed, direction of car, intended distance from the car in front/back, and lateral position within the road. In the case of an experienced driver these actions are predominantly a matter of physical reflex without high-level conscious attention.

In the following, these hierarchical ECOM levels will serve as an anchor for building up a symbolic logical model of human intentions mapping *percepts* onto *actions*. It is thus a symbol grounding problem to link the three protocol-based levels of the *a priori* ECOM model (supplemented by the legally-specified highway driving rules) to the lower-level 'tracking' information provided to us by ground-truth annotation of visual primitives observable through the car windscreen in conjunction with captured *in situ* eye movements and control inputs from the driver. In the most typical mode of operation, the system would construct a logically consistent world-model from the computer-vision systems input and use the conditional dependences from the drivers gaze, signal and control inputs to determine the operating intention and sub-intention of the ECOM model at any given time. The clausal form of the ECOM hierarchy (refer to Fig. 16) shows a flavour of this implementation. It consists of a head clause indicating primary ECOM intention, and a body clause indicating conjunctions of antecedent ECOM intentions and also perceptual conditions that must be fulfilled. Thus the problem of classifying ECOM based (driver) intentions requires the use of rule-based classification algorithms (i.e., allowing the use of logical clause resolution), discussed in the following section.

C. Logical Techniques Appropriate for the Implementation of Protocol-based Task Subsumption Models

Our problem is thus one of classifying (driver) intentions with respect to the ECOM model and *a priori* (highway-code relevant) driving protocols using driver gaze and control inputs. We must thus link protocol-based structures (describable in terms of first-order logic) to low-level features. Although several strategies are apparent in machine learning for achieving this, we found the following techniques quite effective in human intentional classification.

1) First-order logic: In the current work a generative logic module relating to driver knowledge will be implemented declaratively by first-order deductive resolution in *Prolog*, with rules and clauses based on the UK Highway Code in addition to the ECOM model. Prolog evaluates first-order logic clauses by assuming that the negation of the conclusion follows from the premises via the use of a resolution theorem prover acting on *horn clauses* (i.e., proof by refutation). Resolution theorem-proving, however, is error intolerant (i.e., a single contradiction among the predicates allows any proposition to be provable, potentially limiting its application when predication is supplied by potentially fallible detectors (the extent to which this applies in our case, with predication supplied by ground truth computer vision primitives and control inputs, is thus a key subject of evaluation).

2) Decision trees: We also evaluate a 'zeroth' order logical induction algorithm in the form of the (Quinlan's) Decision Tree induction algorithm (inductive inference can be considered the inverse of deduction), the use of which is advantageous in many cases; they are fast classifiers that discretise the decision space so as to explicitly avoid overlaps of classification regions, with classification of data performed in a progressive hierarchical manner, partitioning the feature space recursively into hyper-rectangles. Decision tree algorithm can potentially result in suboptimal tree structures due to over-fitting of training data, nevertheless it generates rules that are simple, interpretable and accurate in many different applications, even in cases where training data is sparse, or noisy [13].

3) Markov Logic Networks: At the generative level, a state-of-the-art approach for accommodating imprecision in logical clauses is the Markov Logic Networks (MLN) [14]. A Markov Logic Network is an amalgam of *first-order logic* and probabilistic graphical models *Markov Networks* (also known as Markov random fields) that treats first-order logic clauses in *probabilistic* terms. MLNs relax the boundaries of *strict* first-order logic clauses such that, while all unsatisfiable formulas have a zero probability, the set of all entailed formulas have a *maximum* probability of 1 [14]; each logical formula is thus provided with a weight (usually a real number) in relation to a knowledge base of simultaneously asserted predicates. A MLN L is formally defined as a set of pairs (F_i, w_i), where F_i is a first-order logical formula and w_i is a real number that defines the weight of F_i. Given a finite set of constants C, a Markov network $M_{L,C}$ consists of a single binary node for each possible grounding of each predicate appearing in L, and one grounded feature per formula F_i in L, which has a value of 1 for a *true* ground formula, and 0 otherwise. A *Ground Markov Network* is one in which vertices are ground atoms (*ground predicates*) that can be collectively associated with a concrete interpretation (i.e., a possible world). Given a set of ground atoms R defined by predicates in the MLN and the set of constants C, MLN specifies a probability distribution over the set of possible worlds Q (i.e., set of truth values to each ground atom in R) by building a Gibbs measure and partition function as follows [14]:

$$\begin{aligned} P(R = x) &= \frac{1}{Z} \cdot \exp\left(\sum_i w_i \cdot n_i(x)\right) \\ &= \frac{\exp(\sum_i w_i \cdot n_i(x))}{\sum_{x \in Q} \exp(\sum_i w_i \cdot n_i(x))} \end{aligned} \quad (1)$$

where $n_i(x)$ defines the number of true groundings of the ith formula in possible world x.

III. Experimental Objectives

The current work aims to classify human (driver) intentions with respect to *a priori* driving protocols by assuming the existence of a (driver) Perception-Action hierarchy. ECOM is used for collating human driving strategies in formalised, protocol-expressible-terms (i.e., first-order logical rules), in terms of visually-perceivable entities of the appropriate hierarchical level (e.g., lane boundaries, traffic-light states). The classification system must thus address the symbol grounding problem [8] involved in linking the *a priori* ECOM intentions to low-level features such as computer vision, eye gaze, and control inputs. We examine two principle experimental models, the generative and the discriminative, in seeking to reconcile the logical nature of the ECOM intentions with low-level feature input. We use *Prolog* as an explicitly generative first-order logical modelling system and decision trees as the discriminative logical modelling system. We compare these with Markov Logic Networks (in a variety of learning configurations), as an alternative method for reconciling first-order logic with uncertain predication. We finally evaluate a hybrid framework for combining stochastic decision-tree learning with the generative structure of first-order resolution theorem proving in *Prolog*.

IV. Experimental Data Collection

A. Experimental Dataset

1) Recording: Training and test data were recorded (and provided) by Autoliv Development AB, (Sverige) using a sensor-equipped vehicle driven around Stockholm, (Sweden), by a single driver in the absence of additional passengers, driver instructions and navigation equipment (route planning), with least biasing driver assumptions (e.g., *'drive around town'*). The collected data consists of low level features including eye-gaze location (captured from an array of eye-trackers with Gaze angle, Head rotation of $\pm 110°$, and Head position with millimetre precision), control features (i.e., steering angle, braking and acceleration), external video scene capture using three $180°$ panoramic view cameras, (20 Hz sweep) LIDAR, and (20 Hz) DGPS coordinates of the experimental vehicle.

2) Contextual Labels: The original dataset comprises a total of 158,668 frames, out of which, 47,923 frames cover non-urban, 82,424 frames for inner-urban (inner city locations), and 28,424 frames cover outer-urban environmental locations. There are 2,007 frames of roundabouts, 17,366 frames of crossroads, 7,895 frames of T-junctions, 29,865 frames of pedestrian crossings, 31,269 frames of single-lanes, 86,879 frames of double-lanes, 38,880 frames of motorways, 21,799 frames of traffic-lights, 6,462 frames of road-markers, and 3,387 frames of traffic-road signs.

3) Current Work: The test/training dataset used for current experiments is a subset of the original driving data (discussed above). It consists of six micro-annotated cross-road traversing scenarios, with two cases each of left-turning, right-turning, and straight-over behavioural scenarios, deemed suitable for the current work due to the absence of defects/artefacts in the visual domain with provision of a good data-quality showcase for the *Tracking, Regulating and Monitoring* intentions of the ECOM control model, along with their conditional logic dependences on environmental entities such as traffic lights and signs. They constitute a total of 3278 frames (per frame image size of 244 x 900, at 15fps sampling) of data for which bounding boxes and intentional labelling is obtained. The six scenarios collectively constitute a super-set from which, in principle, all forms of configuration-changing road traversal behaviours can be derived (e.g., T-junction traverses, rotary 'round-about' traverses, etc).

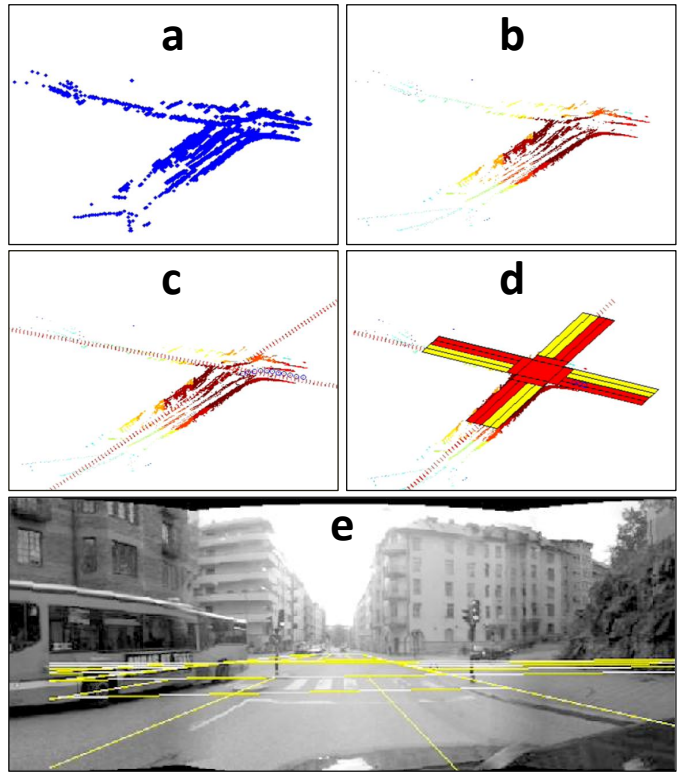

Fig. 2: Five different stages of the propagation of key ground-plane entities utilising the LIDAR data.

B. Ground Truth Annotation of Junction Data

In order to map the ECOM based driver intentions onto the high-way code-relevant entities (i.e., junction entities recorded by external cameras that are deemed relevant to driving protocols), we carry out (per-frame) ground-truthing (hand-labelling of the dataset) at the "Regulating/Monitoring" levels of ECOM (i.e., if a junction objects is present in the recorded visual scene it is asserted as binary *True* and *False* otherwise). The "Tracking" level of ECOM is not itself directly annotatable and so does not constitute part of the classification problem; however, due to its hierarchical linkage with the other levels, it still has the potential to influence classification by virtue of its influence on the control features (e.g., steering, braking, indicators and acceleration). In order to collate visual entities (asserted via visual scenes inspection) with cognitive (driver) intentions (deduced via control and signal inputs, as well as gaze behaviour) in a manner directly congruent with the ECOM intention/control model; it is important to to distinguish between 'ground-plane' (i.e., objects defining

junction topology e.g., lanes, roads, pedestrian crossings etc.,) and 'view-plane' (e.g., traffic signs, lights) high-level scene objects, since the visible presence of certain scene objects (e.g., traffic signs or lights) is more important than their geometric position or orientation. For the classification problem of determining (ECOM based) driver intentions, we characterise gaze behaviour, on a per-frame basis, via bounding boxes (placed around junction scene objects manually) encompassing these junction entities. To establish correlations between these 'ground-plane' objects with the (driver's) gaze position it is necessary to have an autonomous propagation of these entities throughout the video footage. This is carried out as follows.

1) Projective Ground-Plane Tracking: The propagation of key ground-plane entities utilises the LIDAR data, and involves five different stages:

a) *Temporal aggregation of LIDAR data to give an approximate delineation of junction-outlines (Fig. 2-a).*

b) *Histogram and drift correct aggregate LIDAR data to further distinguish road outlines and differentiate it from traffic-trajectory noise (Fig. 2-b).*

c) *Canny edge detection is applied to the aggregate LIDAR data [15] and a Hough transform $H(r,\theta)$ is then computed via the edge point mapping: $r(\theta) = x_0 \cdot \cos\theta + y_0 \cdot \sin\theta$, where r, θ are the Hough transform parameters [16], ('r' is the distance between the line and the origin, while 'θ' is the angle of the vector from the origin to the closest point ((x_0, y_0))). A Hough Transform histogram with high angular suppression is used to obtain predominating road vectors, i.e., we obtain a Canny edge detected image such that non-zero intensity values with coordinates (x_0, y_0) in the image plane constitute the Hough intensity $H(r,\theta)$. A selection criterion is applied to the peaks in $H(r,\theta)$ to identify the top two line candidates that are $> 30°$ apart in the θ ordinal i.e., (Fig. 2-c)*

$$\{(r_1,\theta_1),(r_2,\theta_2)\} : \underset{r_1,\theta_1,r_2,\theta_2}{\operatorname{argmax}} \; H(r_1,\theta_1) + H(r_2,\theta_2) \; s.t. \; |\theta_1 - \theta_2| > 30° \quad (2)$$

d) *A junction topology and pedestrian-crossing/lane structure is fitted to $\{(r_1,\theta_1),(r_2,\theta_2)\}$ on the basis of* a priori *knowledge of their absolute number (Fig. 2-d).*

e) *An approximate view-plane transformation matrix is applied for projecting the junction topology into screen frame for small-scale manual adjustment of car-height/camera orientation etc., (Fig. 2-e).*

The outputs of this process are the per-frame gaze occupancies of the projected junction-plane bounding boxes on the driver's view plane, supplemented by the per-frame gaze occupancies of the image-plane objects. (This dataset is available at: http://www.diplecs.eu/data/dataset_ecom.zip/view).

2) Expert Annotation of Intentional States: ECOM levels consists of mutually exclusive intentional classes; however, different levels may be simultaneously active (this form of hierarchical relation is evident to a certain extent by the implicit hierarchical structure of the highway-code-relevant entities e.g., junction $\rightarrow$ road $\rightarrow$ lane). For each of these six scenarios, per-frame driver's visual scene annotation (as well first-order logical clause formation) is carried out by four individuals separately (comprising two psychologists from the

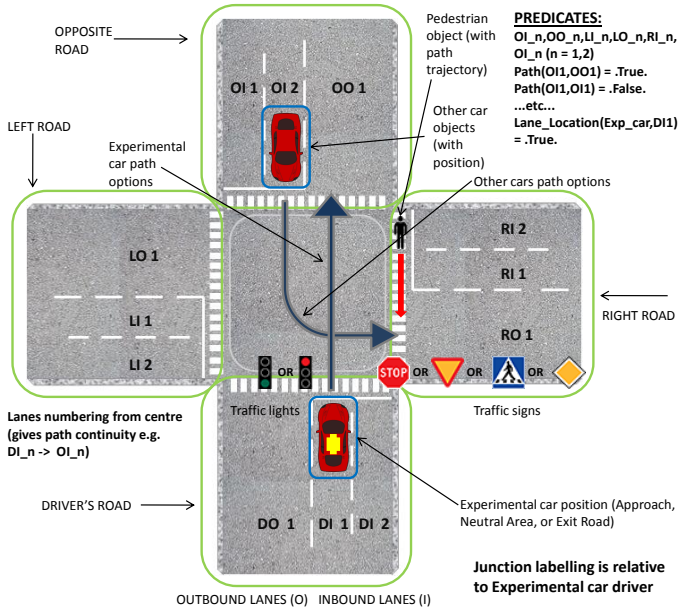

Fig. 3: Cross-Road Junction predicates.

Crisis and Risk Research Centre, MINES ParisTech, and two with engineering expertise) in terms of the ECOM behaviours that are observed (refer to Fig. 4 for full list of annotation states). Because of the size limitation on the actual dataset the over all variation is negligible. In case of a larger dataset the annotation process might spread over a larger number of individuals, increasing the chances of substantial variability over individual annotations (as such, inter-rater reliability study might become inevitable) or an algorithmic annotation and logic-clause formation might prove to be useful. In conjunction with the annotation gaze behavior with respect to bounding boxes of key objects for each frame, this labelling serves as a coarse-grained characterisation of the driver's behaviour on a per-frame basis using high-level entities deemed relevant by the highway code (i.e., traffic lights, lights states, pedestrians, cars, traffic signs, lanes, road-dividers etc). Lower level features are provided by the control inputs and raw gaze positions. The total data set thus consists of *High Level Features + Low-level features*, along with the labels *First Intentional Level state, Second Intentional Level state, ..., Fifth Intentional Level state*. The classification problem is thus one of mapping the high and low level features onto the (ECOM) labels.

In building the classification data-set, all the relevant hierarchical relations are included as far as possible in the feature space (Fig. 3). This allows the stochastic pattern-recognition approach of decision trees a feature space potentially rich enough to mimic the deductive potential of generative first-order logic approaches. We thus expand the initial set of bounding box gaze occupancies so that a full hierarchy of binary features is generated, consisting of junction, road, and lane bounding boxes (i.e., such that the notion of subsumption is implicit within the hierarchy). Thus, lanes at a junction are characterised as belonging to the set $\{ROn, RIn, LOn, LIn, DOn, DIn, OOn, OIn\}$, where n is a number between 1 and the total number of lanes

Driver Intention Hierarchy	ECOM Hierarchy
Level-1 • Navigate Junction	***TARGETING***
Level-2 • Turn Left • Turn Right • Go Straight On	***MONITORING***
Level-3 • Junction Approach • Traverse Neutral Area • Exit Junction	***MONITORING***
Level-4 • Wait for Light to Change Green • Slow for Red Light • Wait for Clear Exit Trajectory • Approach Junction without Stopping • Continue Traverse	***REGULATING***
Level-5 • Assess Traffic while Stopped at Lights • Assess Lights • Assess Exit while Stopped • Follow Car in Front • Assess Traffic while Traversing • Assess Traffic while Exiting • Assess Traffic while Approaching • Check Exit while Moving • Check Lights	***TRACKING***

Fig. 4: Hierarchical levels of intentional annotation states.

of the road (R=right, L=left, D=driver's side, O=opposite side; I/O = inbound/outbound lane). Consequently road *sides* (i.e., inbound/outbound sides of the road) are characterised as belonging to the set: $\{RO, RI, LO, LI,\ DO, DI, OO, OI\}$, with roads as a whole belonging to the set: $\{R, L, D, O\}$ (thus, in general, subsumptive relations are manifested via ordinal subset relations). We also include generalized velocity descriptors such as 'driver-ward', 'left-ward', etc., that subsume the tracking-level orientation-based descriptors, allowing for the possibility of more coarse-grained velocity relations to be captured by the classification process. Thus the complete set of features (i.e., a feature vector) comprise 594 descriptors for each frame of data, with the per-frame intentional annotations constituting the classes to be learned.

V. Human-Intention Classification Strategies

A. Decision Tree Learning

The ECOM hierarchy is defined such that individual intentions are mutually temporally exclusive; individual levels, however, are simultaneously operative. Formally, it can be stated that the classification of ECOM intentions is the simultaneous categorization of the unique item i^l within each level (given a feature vector X); i.e., it is a mapping problem of the form:

$$\forall l, X \rightarrow i^l : i^l = \underset{j^l}{\operatorname{argmax}} \{p(j^l|X)\} \tag{3}$$

A decision-tree learning algorithm based on *Gini impurity* is used for classification on the basis of its readily-interpretable results (rule induction using decision tree algorithms [17], has the characteristic of direct translatability into logical clauses). Gini impurity measures the degree of impurity in a given

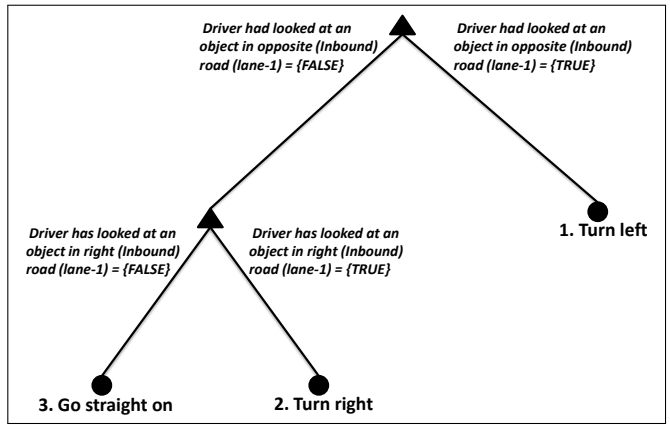

Fig. 5: Decision tree generated for *ECOM* level 2.

dataset comprising multiple class labels, i.e., it measures the probability of a randomly chosen element to be incorrectly labelled given a subset of randomly distributed class labels.

Given an observation dataset and the associated class labels, decision trees are learned by binary recursive partitioning of the sample space into nodes (i.e., features) that terminate on leaves (i.e., class labels). The choice of the best split is based on the smallest impurity criterion (among all possible predictors) by computing the *Information gain* from parent nodes to child nodes using an impurity measure, e.g., *Gini impurity*. In general, if q_i is the frequency of class label i (ECOM intention) in the observation dataset D (training set), also known as the parent set, then for n class labels the Gini impurity $G^D(q)$ is given by:

$$G^D(q) = \sum_{i=1}^{n} q_i(1-q_i) = \sum_{i=1}^{n} q_i - \sum_{i=1}^{n} q_i^2 = 1 - \sum_{i=1}^{n} q_i^2 \tag{4}$$

The observation dataset is further partitioned according to the values of each specific feature x where $x = \{0, 1\}$ within the feature vector X and *Gini impurity* G^{S_x} is computed for each subset S_x. *Information gain* $I(x)$ is computed as impurity degrees of the parent set D and weighted summation of impurity degrees of the subsets $(S_{x=0}, S_{x=1})$. The weight is based on the frequency f of each feature value in D.

$$I(x) = G^D(q) - \sum_{x} (f_x G^{S_x}) \tag{5}$$

$I(x)$ is computed for each feature within the parent set, and splits or partitions (i.e., nodes) in the feature vector are iteratively selected on the basis of the next highest $I(x)$, i.e., optimum feature that produces maximum information gain,

$$x^* = \underset{x}{\operatorname{argmax}} \{I(x)\} \tag{6}$$

In the current evaluation trees are learned and tested using leave-one-out cross-validation. Trees generated by this process are used to classify ECOM intentional levels for each scenario with results as detailed in section VI (see Fig. 5 for an example of generated tree, ECOM level 2 is shown due to its relatively simple node structure). Level 1 intentions are also omitted entirely from the evaluation in view of their non-discriminative nature in junction environments. It is

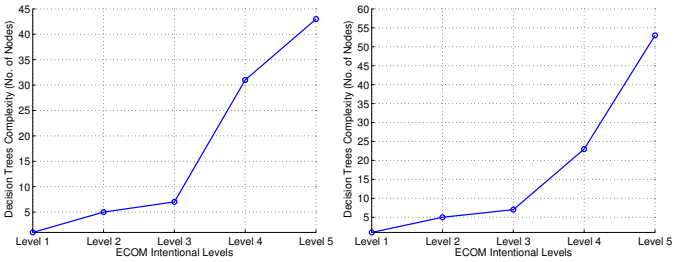

Fig. 6: Decision tree complexity verses ECOM level number for two different training sets.

noteworthy that they depend only on a small fraction of the total set of hierarchical scene descriptors (i.e., features set), with very little compromise on classification performance (in the sense of Gigerenzer's decision making heuristics [18]). An observable characteristic is that decision trees become more complex with increasing ECOM level due to the hierarchical introduction of additional context information (refer to Fig. 6).

B. First Order Deductive Logic

Both ECOM and the legally-mandated rules of the road are essentially protocol-based in nature. This lends them to being rendered as a set of first-order logical clauses, which may be queried by resolution theorem proving with respect to a particular first-order formula. In functional terms, the deductive logic attempts constructs a logically-consistent model of the active ECOM intention/sub-intentions from the computer-vision system, driver's gaze, signal and control inputs. Like ECOM, the deductive logic system embodies the principle of *action* preceding *perception* common to all perception-action systems. This can be formalised into a mathematical principle of *bijectivity* such that:

$$\{\forall_{mn} P^l_m, P^l_n : \exists A(P^l_m, P^l_n)\} \tag{7}$$

i.e., every given ordered percept pair (P^l_m, P^l_n) for level l is linked by an action, or a sequence of actions, A. Thus, if $i^l_{present}$ is the current *ECOM* intention and $P^l_{present}$ is the currently perceived percept, then $A^l_{present}$ is the current *ECOM* action if $i^l_{present}$ defines the following mapping:

$$i^l_{present} : (P^l_{past}, P^l_{present}) \longmapsto A^l_{present} \tag{8}$$

This serves to eliminate redundancy in perceptual predication, while ensuring sufficient descriptive richness to characterise ECOM intentions on a level-appropriate basis[1]).

The logic system employs first-order predicate logic with *a priori* logical predication applied in a top-down manner, starting with the most general world predicates and clauses, such that lower-level predicates add precision to higher level predicates (for example, a specific *road* has associated with it specific *lanes* and so on). Thus predicates are defined in such a way that the subsumptive hierarchy implicit within the ECOM persists between different levels of the logic. The deductive system itself is coded in *SWI Prolog* with a recursive clause structuring where relevant (for example, to define lane adjacency relations). The Prolog-based system carries out deductive resolution in order to generate the the active ECOM intentions and sub-intentions (at the Monitoring/Regulating level) for a given range of frames characterised by the predicatised input features. Assuming the perception-action bijectivity relation given above, intentional clauses thus take either the form (given that $(\phi^t_1, \phi^t_2, ..., \phi^t_n \mid t \in \{past, present\})$ are feature predicates, and we define the binary operator $\Omega \in \{\wedge, \vee\}$):

$$\left(\Omega^n_{k=1}\phi^{past}_k \wedge \Omega^n_{k=1}\phi^{present}_k\right) \Leftrightarrow \textbf{intention} \tag{9}$$

or, if a sub-intention, the form:

$$\left(\Omega^n_{k=1}\phi^{past}_k \wedge \Omega^n_{k=1}\phi^{present}_k \wedge \textit{intention}\right) \Leftrightarrow \textbf{sub\_intention} \tag{10}$$

A flavour of the ECOM-based intentional clauses, in *conjunctive normal form*, is given below for the first two ECOM levels (we omit pathing predicates and predicates directly indicative of intention e.g., signalling): (**Note:** due to space constraints only a few of these intentional clauses are listed).

ECOM Level-1 Intentions:

$\mathcal{R}^1_1$: (¬car(x,Past) ∨ ¬X_junction(y,Past)
∨ ¬position_at(x,y,Past))
∨ (¬car(x,Present) ∨ ¬X_junction(y,Present)
∨ ¬position_at(x,y,Present))
∨ driver_intention(Navigate_Junction)

ECOM Level-2 Intentions:

$\mathcal{R}^2_1$: (¬car(x,Past) ∨ ¬driver_road(y,Past)
∨ ¬inbound_lane(il,Past)
∨ ¬object_at(x,y,il,Past))
∨ (¬car(x,Present) ∨ ¬left_road(y,Present)
∨ ¬out_bound_lane(ol,Present)
∨ ¬object_at(x,y,ol,Present))
∨ ¬driver_intention(Navigate_Junction)
∨ driver_intention(Turn_Left)

(Contextual complexity increases with increasing depth). The logical clause structure dictates that intra-level intentions are mutually exclusive while inter-level intentions have complex conjunctive/disjunctive relations in consequence of the explicitly subsumptive implementation of the ECOM model. A temporalised closed-world assumption is made, with all active predicates in a frame considered as true (and inactive predicates as false), with previous frame data asserted as *previously* true/false.

The logic system thus acts as a frame intention classifier with each of the intentions i^l on each of the levels, l, queried in turn. Clauses are added to enforce mutual exclusivity amongst intentions on the same level so that, for the intention set $\{j^n\}$ defined $\forall n$ we have:

$$\forall l, i \begin{cases} i^l \rightarrow True : l = n, i = j \\ i^l \rightarrow False : l = n, i \neq j \end{cases} \tag{11}$$

Intentional classes are given an equal weighting in the final output, irrespective of level.

[1]Additional perceptual detail is appended at progressively lower levels of the hierarchy, and only insofar as it relates to sub-intentions - thus if predication were extended to the Tracking level, percepts could potentially include such fine-grained features as those required for determining steering angle via optical flow.

C. Composite System: Combining Generative and Discriminative Classification Systems

Two distinct modes of pattern recognition have been employed in the previous sections, the discriminative (i.e., decision trees) and the generative (i.e., the *a priori* logic model). Since these have differing performance characteristics (see section VI), it is appropriate to consider how an ensemble system [19] would perform. However, since we do not have confidence-based outputs that would enable a simple sum, product or maximum-confidence-based ensemble, we must find an alternative method for dealing with discrepant intentional outputs from the two classifiers.

To do this, we select a 'fall back' classifier to act as the default when there is disagreement between the two (obviously agreement of outputs requires no default behaviour). We notice from the results in section VI that there is a clearly-defined per-level performance dominance by one or other of the classifiers e.g., the logic system performs consistently better at *ECOM level-2* whereas decision trees performs consistently better at *ECOM level-3*. To construct the composite classifier we there utilise this fact to select the default classifier output in the case of discrepancy on the basis of per-level performance. Clearly, this performance cannot be determined with respect to the test set, and so leave-one-out cross-validation within the training set (utilising the natural divisions) is used to approximate this value. The algorithm for generative/discriminative ensemble classification is therefore as follows (with performance as indicated in section VI):

Algorithm 1: Combining *decision tree* and *logic system* binarised intentional outputs.

Given: $\psi^l \equiv$ Classification Performance on Cross Validated Training Data
Terms: p = confidence of intention for a given classifier (logic, decision tree), $i^l(tree), i^l(logic)$ = selected intention for (decision tree, logic system respectively)
Data: *feature vector* X
Input: $\left(i^l(tree), i^l(logic) \mid i^l = \underset{j}{\operatorname{argmax}} \{p(j^l|X)\}\right)$
Result: Composite *ECOM* intention: $i^l(comp)$

foreach *feature vector* X **do**
  if $i^l(tree) == i^l(logic)$ **then**
    $i^l(comp) = i^l$
  else if $i^l(tree) \neq i^l(logic)$ ***AND*** *ECOM-level* $\in \{l \mid \psi^l(logic) > \psi^l(tree)\}$ **then**
    $i^l(comp) = i^l(logic)$
  else if $i^l(tree) \neq i^l(logic)$ ***AND*** *ECOM-level* $\in \{l \mid \psi^l(tree) > \psi^l(logic)\}$ **then**
    $i^l(comp) = i^l(tree)$

By combining the decision-tree outputs with logical deduction in this way, the accuracy of the composite system is very significantly greater than that of the individual systems.

D. Markov Logic Networks

In the following, we employ Washington University's Alchemy system (http://alchemy.cs.washington.edu/) for MLN training. This permits MLNs to be employed in two distinct learning modes: clause weight-learning (where pre-existing clauses are weighted on the basis of evidence) and clause induction (in which both clauses and weights are learned). We explore both of these possibilities. Clause-weighting is carried-out with respect to the body of Prolog clauses used earlier (with only syntactic modifications).

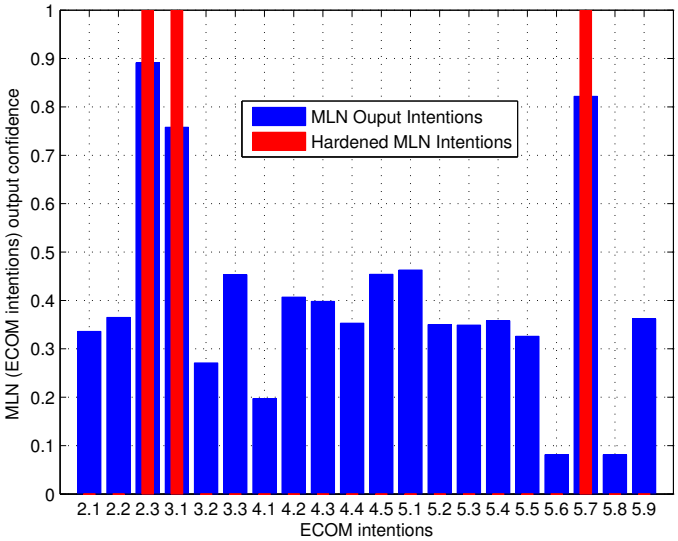

Fig. 7: Comparison of $i^l(mln)'$ against $i^l(mln)$ for (2:5 levels) of ECOM intentions, where (2.1, 2.2 etc.,) represent *(ECOM level 2 intention 1, level 2 intention 2 etc., respectively).*

A generative (rather than discriminative) mode of weight learning is used. Each frame of data (represented as a feature vector) is converted into feature predicates and added to a relational database that acts as a training dataset. We follow an implicit closed-world assumption (i.e., any ground predicates not included in the training datasets are considered as *False*). The MLN is built over clauses defined by the *conjunctions/disjunctions* of *literals* (these define the edges and nodes of the MLN, respectively). The MLN training process commences with a conversion of logical clauses from *first-order logical* format to CNF (*Conjunctive Normal Form*). A weight is then generatively learned for each clause via a maximum likelihood gradient descent of Equation 1 with respect to a relational database (Box-constrained Limited-Memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) method [20] is used here to attain the minimum). The relational database is the training dataset comprising ground atoms or *ground feature predicates*. MLN inference is then used to infer the most likely ECOM intention (given a knowledge base and query formula).

We tested different MLN configurations with a performance criterion given by the accuracy of induction over the whole ECOM intentional hierarchy. Classification outputs from MLN inference are hardened to unity, while nonetheless allowing for the possibility of ambiguity (i.e., so that all greater than than 0.5 confidences are equally represented). Thus, while the inferred MLN output obeys: $(i^l(mln) \in [0,1])$, the hardened MLN output is constrained such that: $(i^l(mln)' \in \{0,1\})$ if (refer to Fig. 7):

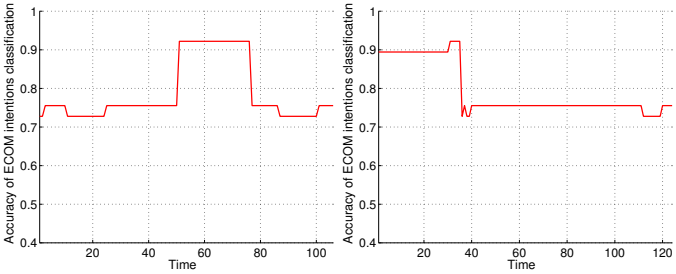

Fig. 8: *Decision trees* classification accuracy plotted over a temporal frame axis for: *Right turning* scenario I (Left), *Straight on* scenario II (Right).

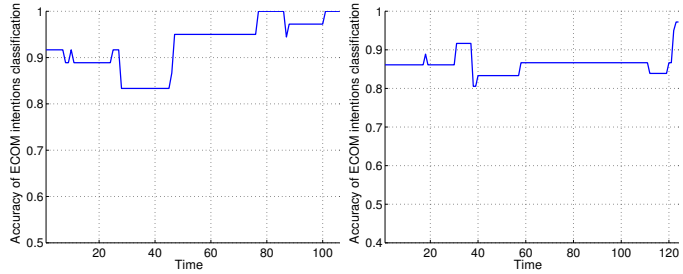

Fig. 9: *First-order logical* classification accuracy plotted over a temporal frame axis for: *Right turning* scenario I (Left), *Straight on* scenario II (Right).

$$i^l(mln)' = \begin{cases} 0 & \text{if } i^l(mln) \text{ conf} \leq 0.5 \\ 1 & \text{if } i^l(mln) \text{ conf} \geq 0.5 \end{cases}$$

VI. Experimental Evaluation and Results

Classification accuracy is found using the *inner product* binary similarity and distance measure [21]. If I^l is the total number of ECOM intentions within a specific level l, L is the cardinality of set comprising all ECOM levels, α is a binary class label representing an intention output of the classifier; $\alpha \in \{0, 1\}$, and β is a binary class label representing an intention output in ground-truth data; $\beta \in \{0, 1\}$, then:

$$\begin{aligned} \gamma^l &= \sum_{i=1}^{I^l} (\alpha_i^l \oplus \beta_i^l | \alpha_i^l = 1, \beta_i^l = 1, l \in L) \\ \nu^l &= \sum_{i=1}^{I^l} (\alpha_i^l \oplus \beta_i^l | \alpha_i^l = 0, \beta_i^l = 0, l \in L) \end{aligned} \quad (12)$$

γ expresses the sum of all cases where α and β are both 1, ν expresses the sum of all cases where the values of α and β are both 0. The *inner product* similarity and distance measure is given as: $\zeta^l = \gamma^l + \nu^l, l \in L$, and the normalised accuracy measure for a specific feature vector X is given as,

$$acc_X^l = \frac{\zeta_X^l}{I_X^l}, l \in L \quad (13)$$

If N is the total number of feature vectors for a specific driving scenario, we compute an average classification accuracy measure for a specific level l as:

$$acc_{avg}^l = \frac{\sum_{X=1}^{N} acc_X^l}{N} \quad (14)$$

In the following, all reported results are obtained from leave-one-out cross-validation, exploiting the natural training set divisions (2 left, 2 right, 2 straight-on junction traverses).

A. Decision Trees Results

Decision Tree classification accuracy for all six scenarios and intentional levels is shown in Table I. Note that *Level 1* has been omitted, since it comprises only one intention (i.e. Navigate Junction, cf IV-B) which is universally active across all junction navigation scenarios. Fig. 8 illustrates the variation of *decision tree* classification accuracy (averaging all *ECOM* intentional levels) over the temporal axis for two typical scenarios.

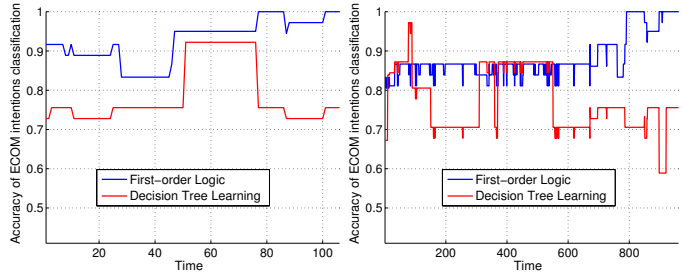

Fig. 10: Comparison of *First-order logical* against *Decision trees* intentional outputs for *Right turning* scenario I (left), *Right turning* scenario II (right).

TABLE I: Classification accuracy using decision tree learning.

Driving Scenarios	ECOM Intentional levels				
	Level 2	*Level 3*	*Level 4*	*Level5*	*Mean*
Right Turn I	0.6855	0.8113	0.8000	0.8574	0.7886
Right Turn II	0.5491	0.9788	0.6789	0.8777	0.7711
Left Turn I	1.0000	0.9991	0.8320	0.8953	0.9316
Left Turn II	0.3333	0.9803	0.7922	0.8830	0.7472
Straight On I	0.3333	1.0000	0.7752	0.8367	0.7363
Straight On II	0.5215	1.0000	0.8000	0.8522	0.7934
Mean	0.5705	0.9616	0.7797	0.8670	

B. First Order Logic Results

Classification accuracy for the logic system is measured across all driving scenarios via Equation 14 using leave-one-out cross validation (see Table II).

TABLE II: Classification accuracy using First-order logic deduction.

Driving Scenarios	ECOM Intentional levels				
	Level 2	*Level 3*	*Level 4*	*Level5*	*Mean*
Right Turn I	0.8553	0.9434	0.9415	0.9675	0.9269
Right Turn II	0.7260	0.9879	0.8541	0.9646	0.8831
Left Turn I	0.9134	0.9720	0.7470	0.9006	0.8832
Left Turn II	0.6667	0.6858	0.9578	0.9819	0.8230
Straight On I	0.6888	0.8367	0.9401	0.9234	0.8473
Straight On II	0.6747	0.9462	0.8952	0.9364	0.8631
Mean	0.7541	0.8953	0.8893	0.9457	

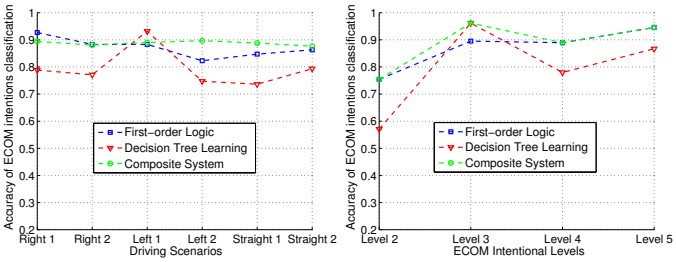

Fig. 11: Comparison of average classification accuracy for *first-order logic*, *decision tree learning* and *composite system*.

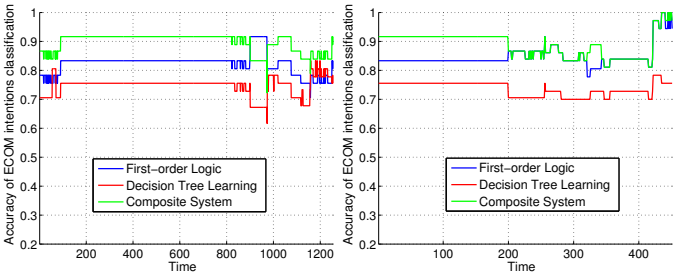

Fig. 12: Comparison of classification accuracy for *first-order logic*, *decision tree learning* and *composite system* intentional outputs over a temporal frame axis for: (*Left turning* scenario II) (left), (*Straight on* scenario I) (right).

Classification accuracy plotted with respect to the temporal frame axis for the *logic system* intentional output is shown in Fig. 9 for two typical driving scenarios. It may be noticed (Fig. 10) that first-order logic accuracy figures increase with time for turning scenarios (while the decision tree output does not), since the default 'straight on' assumption becomes falsified as more temporal context is accrued. This illustrates the distinct advantages of the two methods; the logic system effectively utilises temporal context in *a priori* specified features, while decision trees have the potential to isolate instantaneous discriminators in subsidiary features.

C. Composite System Results

Using leave-one-out cross validation, the classification accuracy for the composite system is determined (Table III).

TABLE III: Classification accuracy for (Decision tree and Logic) composite system.

Driving Scenarios	ECOM Intentional levels *Level 2*	*Level 3*	*Level 4*	*Level5*	*Mean*
Right Turn I	0.8553	0.8113	0.9415	0.9675	0.8939
Right Turn II	0.7260	0.9788	0.8541	0.9646	0.8809
Left Turn I	0.9134	0.9991	0.7470	0.9006	0.8900
Left Turn II	0.6667	0.9803	0.9578	0.9819	0.8967
Straight On I	0.6888	1.0000	0.9401	0.9234	0.8881
Straight On II	0.6747	1.0000	0.8952	0.9364	0.8766
Mean	0.7541	0.9616	0.8893	0.9457	

Fig. 11 and 12 demonstrate an increased classification accuracy for the *composite system*. This illustrates the advantages to be gained from ensembles of the two distinct intentional classification methods (i.e., the generative and the discriminative).

D. Markov Logic Networks Results

As well as clause and clause-weight learning we also consider independent weighting of clauses for different temporal variable instantiations of the MLN (i.e., so that some degree of 'temporal lag' in intentional manifestation can be accommodated). Results for various configurations are as follows:

1) MLN with clause weight-learning & intentional querying of full ECOM hierarchy (without per-state temporal logic learning) -see Table IV.
2) MLN with clause weight-learning & intentional querying of full ECOM hierarchy (incorporating temporal logic learning), - see Table V.
3) MLN *structure* learning followed by intentional querying (MLN logical inference) of the full ECOM hierarchy without any *a priori* first-order clause structures -see Table VI.
4) MLN structure learning followed by MLN weight Learning. The classification performance shown in Table VII indicates significant improvement from post-weighting of induced clauses.
5) MLN with Structure Learning after having been seeded with *first-order logic* clauses used in Prolog tests (with no per-state temporal logic learning), -see Table VIII.
6) MLN structure learning seeded with *first-order logic* clauses and *reweighted* using weight learning -see Table IX.
7) MLN Structure Learning seeded with *first-order logic* clauses (with temporal logic learning) -see Table X.
8) MLN with Structure and Weight Learning after seeding with Prolog clauses (and including temporal logic learning) -see Table XI.

TABLE IV: Classification accuracy for MLN (*Weight-Learning*).

Driving Scenarios	ECOM Intentional levels *Level 2*	*Level 3*	*Level 4*	*Level5*	*Mean*
Right Turn I	0.5409	0.9465	0.8755	0.9612	0.8310
Right Turn II	0.4051	0.9341	0.8293	0.9629	0.7829
Left Turn I	0.6334	0.7962	0.3591	0.7180	0.6267
Left Turn II	0.4090	0.6794	0.9253	0.9789	0.7482
Straight On I	0.8123	0.7953	0.7601	0.8840	0.8129
Straight On II	1.0000	0.8763	0.4290	0.7867	0.7730
Mean	0.6334	0.8380	0.6964	0.8819	

TABLE V: Classification accuracy for MLN (*Weight-Learning comprising temporal logic learning*).

Driving Scenarios	ECOM Intentional levels *Level 2*	*Level 3*	*Level 4*	*Level5*	*Mean*
Right Turn I	0.5409	0.8019	0.9943	0.9686	0.8264
Right Turn II	0.4051	0.8809	0.8398	0.9748	0.7251
Left Turn I	0.6334	0.6623	0.9039	0.8997	0.7748
Left Turn II	0.4090	0.6571	0.9495	0.9839	0.7499
Straight On I	0.8123	0.6312	0.9752	0.9414	0.8400
Straight On II	1.0000	0.5833	0.9952	0.9633	0.8854
Mean	0.6334	0.6694	0.9430	0.9553	

TABLE VI: Classification accuracy for MLN (*Structure Learning without first-order logical clauses*).

Driving Scenarios	ECOM Intentional levels *Level 2*	*Level 3*	*Level 4*	*Level5*	*Mean*
Right Turn I	0.6667	0.6667	1.0000	0.9686	0.8255
Right Turn II	0.6306	0.6254	0.7669	0.9009	0.7310
Left Turn I	0.6667	0.6658	0.9039	0.8997	0.7840
Left Turn II	0.6696	0.6754	0.9402	0.9663	0.8129
Straight On I	0.6534	0.6534	0.9512	0.9015	0.7898
Straight On II	0.3468	0.3468	0.0403	0.0681	0.2005
Mean	0.6056	0.6056	0.7671	0.7842	

TABLE VII: Classification accuracy for MLN (*Structure Learning followed by weight learning*).

Driving Scenarios	ECOM Intentional levels *Level 2*	*Level 3*	*Level 4*	*Level5*	*Mean*
Right Turn I	0.6667	0.6667	1.0000	0.9686	0.8255
Right Turn II	0.6667	0.6719	0.8485	0.9886	0.7939
Left Turn I	0.6667	0.6658	0.9039	0.8997	0.7840
Left Turn II	0.6667	0.6858	0.9578	0.9839	0.8235
Straight On I	0.6667	0.6667	0.9752	0.9414	0.8125
Straight On II	0.6667	0.6667	1.0000	0.9633	0.8241
Mean	0.6667	0.6706	0.9476	0.9576	

TABLE VIII: Classification accuracy for MLN (*Structure Learning seeded with first-order logic clauses*).

Driving Scenarios	ECOM Intentional levels *Level 2*	*Level 3*	*Level 4*	*Level5*	*Mean*
Right Turn I	0.5629	0.7579	1.0000	0.9686	0.8223
Right Turn II	0.6299	0.6250	0.7669	0.9112	0.7333
Left Turn I	0.3333	0.3342	0.0961	0.1677	0.2328
Left Turn II	0.6377	0.6422	0.9482	0.9741	0.8006
Straight On I	0.6962	0.6770	0.8031	0.8620	0.7596
Straight On II	0.3468	0.3468	0.0403	0.2079	0.2354
Mean	0.5345	0.5639	0.6091	0.6819	

TABLE IX: Classification accuracy for MLN (*Structure Learning seeded with first-order logic clauses and Weight Learning*).

Driving Scenarios	ECOM Intentional levels *Level 2*	*Level 3*	*Level 4*	*Level5*	*Mean*
Right Turn I	0.5409	0.7170	0.9887	0.9654	0.8030
Right Turn II	0.6667	0.6719	0.8485	0.9886	0.7939
Left Turn I	0.6072	0.7953	0.2829	0.6320	0.5793
Left Turn II	0.6324	0.6037	0.9382	0.9611	0.7839
Straight On I	0.4339	0.5528	0.2683	0.1693	0.3561
Straight On II	0.3333	0.3333	0.0000	0.1837	0.2126
Mean	0.5357	0.6123	0.5544	0.6500	

TABLE X: Classification accuracy for MLN (*Structure Learning seeded with first-order logic clauses and per temporal logic state learning*).

Driving Scenarios	ECOM Intentional levels *Level 2*	*Level 3*	*Level 4*	*Level5*	*Mean*
Right Turn I	0.6667	0.7610	1.0000	0.9686	0.8491
Right Turn II	0.4436	0.6826	0.8485	0.9886	0.7408
Left Turn I	0.6597	0.6850	0.9039	0.8997	0.7871
Left Turn II	0.6239	0.6048	0.9256	0.9295	0.7709
Straight On I	0.7021	0.6171	0.9525	0.9039	0.7939
Straight On II	0.3468	0.3710	0.0403	0.1577	0.2289
Mean	0.5738	0.6203	0.7785	0.8080	

TABLE XI: Classification accuracy for MLN (*Structure and Weight Learning after seeding with first-order logic and per temporal logic state learning*).

Driving Scenarios	ECOM Intentional levels *Level 2*	*Level 3*	*Level 4*	*Level5*	*Mean*
Right Turn I	0.6635	0.7610	1.0000	0.9686	0.8483
Right Turn II	0.6667	0.6719	0.8485	0.9886	0.7939
Left Turn I	0.6597	0.6850	0.9039	0.8997	0.7871
Left Turn II	0.7023	0.5939	0.9162	0.9341	0.7866
Straight On I	0.6992	0.4967	0.8656	0.7615	0.7058
Straight On II	0.6667	0.6694	1.0000	0.9633	0.8248
Mean	0.6763	0.6463	0.9224	0.9193	

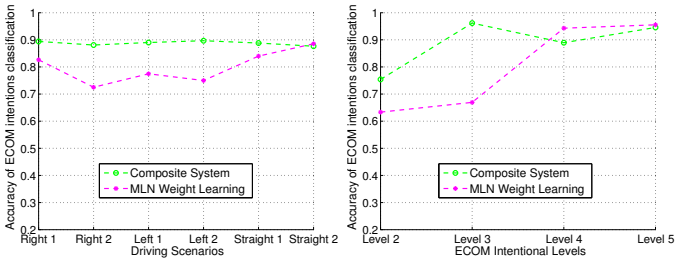

Fig. 13: Comparison of average classification accuracy for *MLN Weight Learning* (configuration-2), and *composite system*.

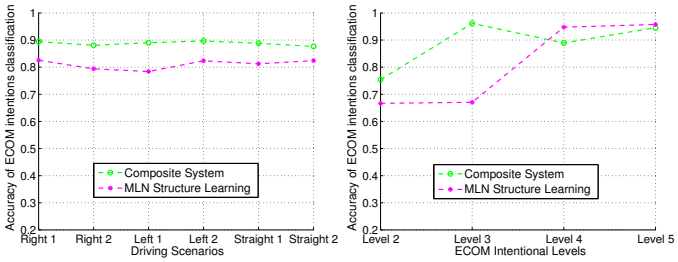

Fig. 14: Comparison of average classification accuracy for *MLN Structure Learning* (configuration-4), and *composite system*.

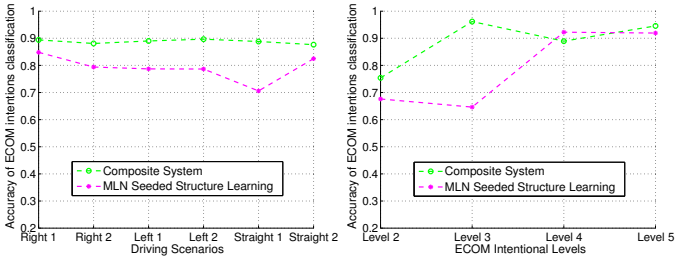

Fig. 15: Comparison of average classification accuracy for *MLN Structure Learning seeded with first-order logic* (configuration-8), and *composite system*.

Results thus indicate that MLN structure learning followed by re-weighting of clauses is the most effective learning configuration; providing additional clauses appears to degrade performance, even when these clauses are effective in the resolution theorem proving context.

E. Comparison of MLNs against (Decision trees & Logic) Composite System

A comparison of MLN classification accuracy against the composite system (see V-C) shows that a hybrid of *generative* (i.e., first-order logic) and *discriminative* (i.e., decision tree) intentional classification systems in fact gives better performance (accross junction navigation scenarios and levels of the ECOM intentional hierarchy). Fig. (13, 14 and 15) plots the classification accuracy for different MLN configurations against the composite system.

VII. Conclusion and Future Work

This paper sought to determine an appropriate supervised methodology for the detection of driver intentions in a manner

that would potentially be of use in cognitive driver assistance systems. To do so, we utilised the psychologically-motivated (*Perception-Action*) model *ECOM* (Extended Control Model), for modelling driver behaviour (capturing the parallel mapping between *task-subsumption* and *scene-representation* hypothesised to exist within (human) drivers as they employ different levels of the P-A hierarchy). An instrumented car traversing a number of different junction navigation scenarios was used to record control inputs with respect to the external driving scene, providing the required experimental dataset. The data directly relates to the *percepts* and *actions* implicit within the ECOM model. We characterised human driver intentions at different levels of the P-A hierarchy via a number of rule-based classification algorithms, i.e., *Markov Logic Networks*, *decision tree learning*, (Prolog-based) *first-order resolution theorem-proving*, and a hybrid (or ensemble) classification system, for which performance results were obtained. It was also established that the hybrid approach outperformed all other classification techniques, and therefore would form an effective basis for cognitive driver-assistance systems based on a P-A model of human intentionality.

The current experimental dataset comprises a single driver throughout the whole sequence, with six junction navigation scenarios. The current size and driver limitations are due to data-quality constraints (for visual footage and sensory data), as well as most importantly the high level requirement of human effort for dense annotation of 594 ECOM-based sensory features and intentional states per frame. In order to expand this model for a practical implimentation in future driver assistance systems, the proof-of-concept evaluations need to be expanded over to larger datasets comprising multiple drivers and noisy features (e.g., erroneous eye-trackers), where human annotation might be replaced by software based auto-annotation. Although the currently in place proof-of-concept system does not completely depend upon the *a priori* ECOM formulations, since decision trees and MLN structure learning intrinsically perform rule induction of intentional clauses, nevertheless replacing Prolog (deductive logic) with Progol (inductive logic) can possibly be useful in that novel rules describing driver intentions can be induced from specific driver behaviours at different levels of the P-A hierarchy, enabling the system to consider the individuality in driver intentional behaviour. Within this context, more generally the advantage of the perception-action approach (as exemplified by the subsumptive bijectivity criterion of Equation 8) is that redundant environmental description is eliminated at each stage of the subsumptive hierarchy of intention. We believe that this is a very generic principle, both in terms of modelling of human intentions, but also in building autonomous cognitive systems for a wide variety of applications.

IMPLEMENTATION OF HIERARCHICAL ECOM INTENTIONAL MODEL					
Task	Environmental Condition	Driver Perceptual Condition	Ordered Sub Tasks	Environmental Condition	Driver Perceptual Condition
Turn Left	At T junction OR at X-Road	Identified Junction	1. Stop	Red light ahead	Spotted red light
			2. Set Indicator	If no In-lane direction signs	No In-lane signs spotted
			3. Do:		
			a) Attain Higher Speed	Sub task 1. has taken place Car has not yet reached speed of any vehicle in front (Tracking sub-sub task)	Intentionally stopped
			b) Turn		Has looked left
			5. Stop	If traversed lane will not be clear during manoeuvre	Has seen car in traversed lane
			6. Attain higher speed	Sub task 5. has taken place AND traversed lane will be clear during manoeuvre	Intentionally stopped
			7. Get in lane	If over threshold of left hand road	
Turn Right	At T junction OR at X-Road	Identified Junction	etc	etc	
etc	etc				

Fig. 16: Illustration of clausal form of ECOM.

ACKNOWLEDGMENT

The work presented here was supported by the the European Union, grant DIPLECS (FP 7 ICT project no. 215078).

REFERENCES

[1] L. Malta, C. Miyajima, and K. Takeda, "A study of driver behavior under potential threats in vehicle traffic," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 10, no. 2, pp. 201 –210, 2009.

[2] V. Cutsuridis, A. Hussain, and J. G. Taylor, *Perception-Action Cycle Models, Architectures, and Hardware*. Springer New York, 2011.

[3] K. Barber, "A feature-based cad representation enabling case-based planning across multiple manufacturing applications," in *SMC, 'Humans, Information and Technology'.,IEEE Int Conf*, vol. 1, oct 1994.

[4] G. Granlund, "Organization of architectures for cognitive vision systems," in *Proceedings of Workshop on Cognitive Vision*, Schloss Dagstuhl, Germany, October 2003.

[5] J. J. Gibson, *The Theory of Affordances*. Lawrence Erlbaum, 1977.

[6] R. A. Brooks, "A robust layered control system for a mobile robot," Cambridge, MA, USA, Tech. Rep., 1985.

[7] D. Windridge and J. Kittler, "Perception-action learning as an epistemologically-consistent model for self-updating cognitive representation," *Adv in Exp Med and Bio, 657:95-134*, 2010.

[8] S. Harnad, "The symbol grounding problem," *Phys. D*, vol. 42, no. 1-3, pp. 335–346, 1990.

[9] M. Shevchenko, D. Windridge, and J. Kittler, "A linear-complexity reparameterisation strategy for the hierarchical bootstrapping of capabilities within perception-action architectures," *Image Vision Comput*, 2009.

[10] E. Hollnagel and D. D. Woods, *Joint Cognitive Systems: Foundations of Cognitive Systems Engineering*. CRC Press, Taylor & Francis Group, Feb 2005, pp. 149–154.

[11] D. Vernon, G. Metta, and G. Sandini, "A survey of artificial cognitive systems: Implications for the autonomous development of mental capabilities in computational agents." *IEEE Trans. Evolutionary Computation*, vol. 11, no. 2, pp. 151–180, 2007.

[12] G. Youngblood and D. Cook, "Data mining for hierarchical model creation," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 37, no. 4, pp. 561–572, July 2007.

[13] W. Duch, R. Setiono, J. M. Zurada, and S. Member, "Computational intelligence methods for rule-based data understanding," in *Proceedings of the IEEE*, 2004, pp. 771–805.

[14] M. Richardson and P. Domingos, "Markov logic networks," *Mach. Learn.*, vol. 62, no. 1-2, pp. 107–136, 2006.

[15] M. Basu, "Gaussian-based edge-detection methods-a survey," *SMC, Part C:, IEEE Transactions on*, vol. 32, no. 3, pp. 252 – 260, aug 2002.

[16] A. Bonci, T. Leo, and S. Longhi, "A bayesian approach to the hough transform for line detection," *SMC, Part A: Systems and Humans, IEEE Transactions on*, vol. 35, no. 6, pp. 945 – 955, nov. 2005.

[17] D. E. Johnson, F. J. Oles, T. Zhang, and T. Goetz, "A decision-tree-based symbolic rule induction system for text categorization," *IBM Systems Journal*, vol. 41, pp. 428–437, 2002.

[18] G. Gigerenzer and W. Gaissmaier, "Heuristic Decision Making," *Annual Review of Psychology*, no. 1, pp. 451–482, 2011.

[19] J. Kittler, M. Hatef, R. Duin, and J. Matas, "On combining classifiers," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 3, pp. 226 –239, mar 1998.

[20] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Math. Program.*, vol. 45, pp. 503–528, December 1989.

[21] S. S. Choi, S. H. Cha, and C. Tappert, "A Survey of Binary Similarity and Distance Measures," *Journal on Systemics, Cybernetics and Informatics*, vol. 8, no. 1, pp. 43–48, 2010.