Inferring Hidden Status and Intents in Video by Causal Reasoning

Amy Fire and Song-Chun Zhu

Abstract—This paper presents a method to infer the hidden statuses of objects or intents of agents (these are called fluents) from video through causal reasoning. Event analysis from video provides detections of status changes (light turn on, turn off) and possible actions, given a number of objects in the scene that we care about (ex: cup, etc). These probabilistic detections, however, may be incorrect or ambiguous. Over time, we use dynamic programming to isolate the best interpretation of the scene, jointly considering detections and causal knowledge. Transition probabilities are dictated by 2 things: 1) the survival analysis histogram (given certain time t duration, how likely the status change during this time duration). 2) if observe a certain action, then how likely change from A to B. if don't observe anything, then this is inertial probability (how things to work). Our modifications to the dynamic programming algorithm accomodate the non-linearity by allowing the insertion of a new action or fluent change.

Our system outputs the hidden status that is not observable in the video, through causal inference. The model incorporates causal reasoning with spatio-temporal detection to generate multiple interpretations of what is transpiring in a scene and to rank those interpretations according to probability. We show that such interpretations can be used to correct (mis)detections of fluents and events in video; results are comparable to humans' performance in reasoning values of hidden fluents.

Index Terms—Causal inference, commonsense reasoning, event analysis, fluents

1 Introduction

RESTRUCTURE INTRODUCTION

.

— PARAGRAPH 1 – general statement about teleological stance and causal effects

Motivation: World/Environment we live in has been designed in a way to reflect cause and effect relationships. Any action is planned to achieve a change in the world: either as a status of object, or the hidden mind.

So we take the teleological stance. action is taken, triggered by certain condition/status of object. at the same time, they are aimed to change the status (of object or the agents mind) of the world. For example, the elevator: push the elevator button, it lights up... Our world was designed in such a way... By dialog too...

Almost every action/event has a goal to change the world. And every action was triggered by condition. However, many of this conditions (object state or fluent/state of the mind) are hidden. So only a portion are observable.

— PARAGRAPH 2 – show some examples – scenario

Focus on examples of drinking (water lower whether cup empty or not) and locked door and approaching car (intent: maybe forgot key).

— PARAGRAPH 3 – explicit/concrete

make explicit: In those examples, fluent triggers action. thirsty triggers getting cup. action triggers fluent change: after drinking water, the cup was empty. get water: becomes full.

and this process includes inference and forward/backward in time....

— PARAGRAPH 4 –

say what is input/output again. In this paper,...

[[FROM NIPS REVIEWS: The general question addressed in the paper is incredibly important (and challenging). The applications of this approach are wide reaching not only for applied questions (automatic recognition of the content of video sequences) but also for theoretical understanding of how humans resolve similar uncertainty in their daily reasoning. The authors nicely position this work along side not only state-of-the-art machine vision systems but also core issues in cognitive science.]]

HUMANS are capable of employing commonsense reasoning to extend inference beyond observation. For example, consider the following office scene:

1) A person sitting behind a computer screen reaches over and moves the mouse, then starts typing. What is the monitor's display status? Is the computer on? Asleep?
2) Later, the person grabs a cup and moves to the water dispenser. Why did he get water? Was he thirsty? Is the cup empty?
3) The agent moves to the wall and raises his arm. The light goes out. Did the agent turn the light off?

The values of the fluents in the first two scenes (i.e., the statuses of the objects, such as whether the monitor dis-

- *The authors are with the Department of Statistics, University of California, Los Angeles, CA 90095.*
  E-mail: amy.fire@ucla.edu, sczhu@stat.ucla.edu

play is active and whether the man is thirsty) are hidden, but an intelligent observer equipped with commonsense causality can answer the posed questions by connecting information. Figure 1 represents one possible inference process in which values of hidden fluents are filled in as preconditions or effects of the observed actions. Even without seeing the screen or the computer, an observer can infer that the agent in the scene moved the mouse to wake the monitor, and that when the typing begins, both the computer and the monitor are likely on. In the second scene, an observer can infer that the agent's actions were triggered by his thirst and the emptiness of the cup.

Missing figure

[[MOVE FIGURE 1 EXPLANATION TO SECTION 1.3 (JOINT INFERENCE)]]

Finally, in the third scene, the fluent value is observable, but the precise action is harder to detect. Without seeing the agent "flipping the light switch", the observer can still reason the agent performed the action based on the observed effects.

FIGURE 1: DITCH HIERARCHY (keep only actions at top). (OBSERVED OR INFERRED). AND FLUENT CHANGE DETECTION–OBSERVABLE FLUENT CHANGE FOR THE SEQUENCE. STILL HAVE TRIGGERS AND CAUSAL EFFECTS. GET RID OF T-AOG.

Inferring fluents of objects and agents together with agent's actions is crucial to understanding that agent's actions, intents, and probable future actions. Therefore, the ability to jointly infer the values of hidden fluents with actions is an important step for artificial-intelligence applications such as constructing situationally aware robots and developing intelligent video-surveillance systems. These reasoning and inference tasks, however, have not been studied in current vision literature. This paper presents methods for jointly inferring values of fluents with agent actions from video input.

1.1 Fluents of objects and fluents of the mind

First introduced by Newton [?], the concept of fluents has been adapted in the commonsense reasoning literature to describe time-varying statuses of objects [?]. Although both describe objects, "fluents" are defined quite differently from "attributes", which have received much attention in recent vision literature (e.g., [?]). Whereas the values of attributes such as the color or texture of an object remain constant over the course of a video, the values of fluents change. The research described in this paper examines two kinds of fluents:

1) Object fluents, e.g., whether a monitor is powered, a light is on, a cup has water, or a door is locked. Because of limitations on visibility and detectability, the values of these fluents are often hidden. They are connected to actions as preconditions or effects.
2) Fluents of the mind, e.g., whether an agent is thirsty, hungry, or tired. An agent's state of mind is completely hidden. Fluents of the mind act as triggers to an agent's actions.

1.2 Fluents over time

[[moved here to FINISH explanation of all fluents (HAD DONE FLUENTS OF OBJECTS, THEN FLUENTS OF MIND, NOW FLUENTS OVER TIME]]

Whereas the work in [?] seeks only a causal explanation at t, it does not consider how the values at each time are connected.

As time increments, $t = 1, 2, \ldots, n$, the fluent F takes on a sequence of values, $F(t) = f_t$, or $(f_1, f_2, \ldots, f_n)$. The values in the sequence, however, are not independent from each other—at least two dependencies exist.

First, the fluent value at $t-1$, together with actions in recent history, restrict the values available for the fluent at t. For example, if the light is on at t and someone touches the light switch, then it is *not* possible that touching the switch changed the light from off to on. This obvious statement forms an important consistency constraint on the sequence. In particular, this constraint allows the model presented in this paper to overcome imperfect spatio-temporal detections from video, providing a coherent sequence of values for the fluent over the course of the video.

The second dependency originates due to the duration a fluent has maintained a particular value without agent intervention. Where some changes in fluent values are attributable to causing actions, other changes occur due to a strong dependence on time. The duration a fluent has maintained a particular value provides information for when a transition to another state will occur. The screensaver on the computer, for example, changes after a pre-specified allotment of time, and this amount of time is usually in 5 minute increments. Agents also experience duration constraints, where, for example, the longer a person goes without drinking, the sooner he will be thirsty.

inlude uniform histogram for fluents that don't have timer mechanism.

EXPLAIN MORE DETAILS ABOUT SURVIVAL ANALYSIS. AND WHY EVERY 5 MINUTES HAVE INSTANCE–BECAUSE MICROSOFT SET IT UP THIS WAY.

BY DEFAULT, HAVE AN INERTIAL PROBABILITY... HOW LONG THINGS WILL NOT CHANGE. TALK ABOUT THIS CONCEPT.

Figure 2 shows some histograms of durations. After removing adjusting for when the screensaver remains

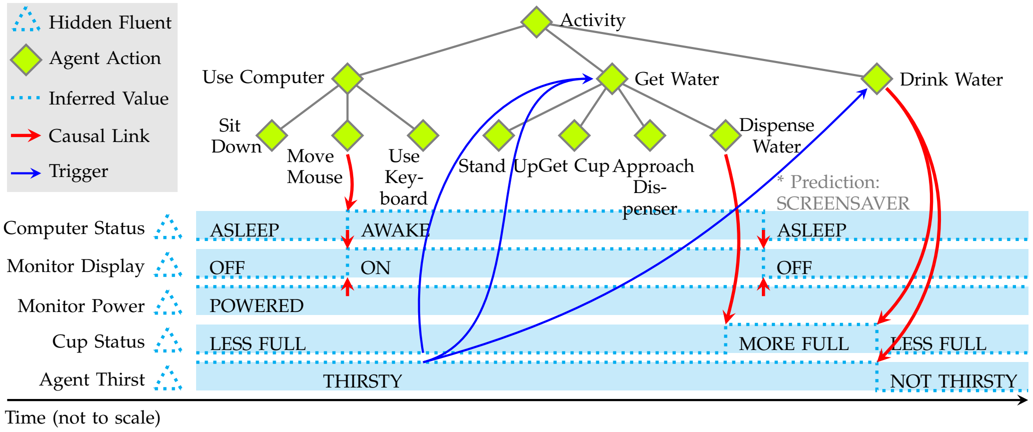


Fig. 1. Illustration of the scenarios from the beginning. Observed actions are used to infer values of hidden fluents, and values of observed fluents are similarly used to infer hidden actions. The inference process extends beyond single instances, providing coherent reasoning solutions both backward and forward in time. TODO: fix word spacing, make fill page width, add turning off light.

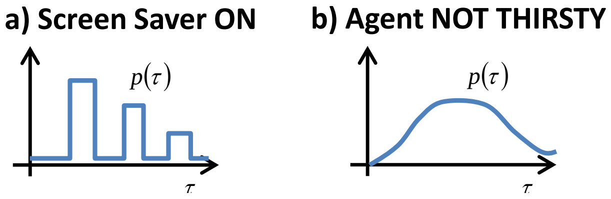

a) Screen Saver ON
b) Agent NOT THIRSTY

Fig. 2. Examples of marginal histograms of durations of F. TODO: rework with tikz/pgf, make thirst flatter

off due to causal actions such as moving the mouse or using the keyboard, the histogram for the duration the screensaver is off, shown in (a), is mostly discrete. The histogram was acquired by asking participants for their screensaver set times. (TODO: experiment) Where commonsense knowledge is available, these histograms can be directly coded. When evidence is available, these distributions can be learned from observation.

For more subjective fluents, such as the thirst of an agent, where limited information is known ahead of time, flatter histograms are used such as that shown in Fig. 2(b). Uniform histograms are used to model fluents where an agent's action removes any time dependence.

The research presented here extends current causal models of fluents to accommodate consistency constraints and probability distributions on durations.

This research builds on recent progress in action and event recognition. Event recognition fills in missed or occluded actions through grammar models [?] or hidden Markov models/dynamic Bayesian networks [?], [?]. To improve recognition rates for small or challenging objects, joint inference is performed over the context, taking agent actions together with the objects or parts [?].

The research presented in this paper extends the trend of using joint spatial, temporal, and causal inference one step further, using detected actions to infer particular values for fluents and using detected fluent values to confirm or correct action detections. These inferred values are then used to reason backward and forward in time, providing coherent values for fluents and actions over the course of the video. For example, an agent filling a cup suggests that the cup lacked water before the action and contains more water after it.

We use the same methods to connect actions to fluents of the mind. Recent cognitive-science literature has focused on goal inference [?], [?], and this framework has been extended to vision research [?]. The goal detection presented in [?] identifies actions as part of a routine, or a larger set of actions, and then information about the routine is used to predict future actions, such as those that would complete the routine. However, this line of research has so far not explored the larger goal of the routine, such as thirst motivating an agent to obtain and drink water. Another contribution of the present study is in connecting the larger routines of actions to their triggering fluents through causal reasoning.

1.3 Joint inference: Action and fluent interaction

Point out: THIS IS 2-WAY INFORMATION–FROM ACTION TO FLUENTS AND SOMETIMES FLUENT TO ACTION

EXPLAIN FIGURE 1 BETTER (MOVE HERE)... USE THIS EXAMPLE TO EXPLAIN HAVE FLUENT CHANGE, OBSERVED ACTIONS. MAKE SOME OF FLUENTS OBSERVABLE (NOT ALL HIDDEN). ADD LIGHT EXAMPLE (INFER SOME ACTIONS: TURN ON LIGHT, WHICH IS NOT OBSERVABLE). SHOWING 2-WAY STREET. ADD DASHED DIAMOND TO THE TOP.

Each of the fluents thus far described takes a value at each instant of time, allowing the notation $F(t)$ to describe the value the fluent F takes at time t. For example, if F represents the light status, then $F(t) = \text{ON}$ indicates that the light is on at time t.

At each instance of time, there is a causal explanation for the fluent value, $F(t)$. This causal explanation was studied in [?], and connects $F(t)$ to actions by attributing the fluent value either to a particular action or to a lack of change-inducing action. Following the light example, the causal explanation could be because an agent just turned the light on or because the light was on and no one turned it off.

The causal connection between values of fluents and agent actions matches the values of fluents and agent actions in a commonsense way. This key connection allows inference of hidden fluents and of actions, even when they are not detectable or when they may be detected incorrectly.

For example, while the action of going to the wall and raising a hand is detectable from video, without labelling the light switch as the object being touched on the wall, it is impossible to tell that the action is "flipping the light switch". However, an understanding of the cause and effect relationship allows the cause ("flipping the light switch") to be probabilistically detected (matched the the detected action) once the effect has been detected (light has come on).

This paper extends methods in [?] to accommodate joint inference, allowing values of fluents to provide feedback in order to inform detections of agent actions.

1.4 Related work

NEED SEPARATE SECTION DEVOTED TO Related Work

1: PSYCHOLOGY IN CAUSAL INFERENCE. KEN NAKAYAMA'S WORK (ILLUSORY CAUSAL CRESCENTS: MISPERCEIVED SPATIAL RELATIONS DUE TO PERCEIVED CAUSALITY)–VISUAL PERCEPTION.

2: COMMONSENSE REASONINING – USING LOGIC TO REASON, BUT THIS IS TOO FRAGILE.

3: INFER CAUSALITY IN STATISTCS– PEARL BAYES NET (NOT CONNECTED/GROUNDED TO IMAGES IN VIDEO.

4: EVENT ANALYSIS – HERE, SAY "WE'RE JUST BUILDING ON THIS ... MOVE SOME OF THE "BACKGROUND" SECTION HERE.

1.5 Summary of contributions

This paper presents a method for joint temporal-causal inference that allows the coherent inference of fluents for the duration of a video.

An overview of the joint inference process is shown in Figure 3. Each video is first temporally described with probability. These descriptions and probabilities are then processed with potential causal explanations, and the best performing causal description is returned. By following the causal description back to its individual components, hidden actions and values of fluents are determined.

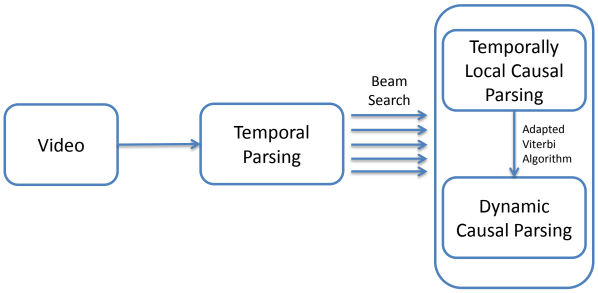


Fig. 3. Joint inference

CLARIFY FIG 3 FOR MY 2 PARTS: (PARSING ON C-AOG AND DP WITH INSERTION FOR REASONING). CLARIFY INPUT: ASSUME OBSERVED FLUENT CHANGE AND ACTIONS INPUT ARE INDEPENDENT (AT EACH TIME) OF EACH OTHER, WITHOUT INVOLVING TEMPORAL AOG. OUTPUT: HIDDEN FLUENT INFERRED... REFER BACK TO FIGURE 1.

We summarize our contributions here:

- Joint inference of fluents and actions. A model is derived that uses causal information to connect fluents to actions consistently over the duration of a video.
- Inference of fluents over time. The joint model integrates constraints to allow consistent inference of fluents over time. Algorithms are provided to navigate the solution space.
- Inference of trigger conditions. The joint model uses trigger conditions, both of states of the world and of the agent's mind, to provide richer reasoning of fluents in the scene.
- Exploration of human cognition of fluents. Experiment results that include human judgements enlighten the community about human processing of fluents.

PARAGRAPH: THE PAPER IS ORGANIZED AS FOLLOWS...

2 Background

Causality is the key connecting actions and fluents (hidden or not) in video. Causality, as studied by commonsense reasoning and AI researchers, is often formulated in terms of first-order logic [?]. However, purely deductive methods typically do not allow for probabilistic solutions, which are important in computer vision research where maintaining ambiguity is essential. While methods exist that allow for probability [?], [?], these methods have not been grounded on vision sensors. While Bayesian networks are commonly used to represent causality [?], the expressive power of a grammar model can represent a greater breadth of possibilities than a single instance of a Bayesian network [?], making it more suitable for vision applications.

Further, grammar representations are advantageous over competing Hidden Markov Models (HMM) and Dynamic Bayesian Networks (DBN) for representing events. By allowing for multiple configurations and high level structures, the hierarchical structure of grammar provides maximum flexibility in representing events in video.

Grammar models are embodied in the graphical representation of the And-Or Graph (AOG) [?]. The AOG was first applied to vision in the spatial domain [?], and has since been extended to the represent temporal explanations [?] and causal relationships [?]. In the AOG, Or-nodes are used to represent alternatives, and And-nodes are used to represent hierarchical decompositions. How these nodes are used depends on the context: temporal or causal.

In this section, we introduce the AOG as used for events and causality in this paper.

2.1 Describing actions: the T-AOG

TALK J(MINIMALLY) ABOUT PARSING, ACTIONS, EVENTS.... JUST HIGHLIGHTING THE INPUT

Many event parsing techniques use a stochastic grammar with some using context-free and others using context-sensitive. All these grammars for event parsing can be embodied in the graphical representation of the Temporal And-Or Graph (T-AOG) [?], a sample of which is shown in Figure **??**.

In the T-AOG, the And-nodes (solid circles) allow for compositional hierarchy, with events decomposed into subevents, such as the decomposition of action A_3 into subactions (a_{31} or a_{32}), and a_{33}. The Or-nodes (dashed circles) provide alternate configurations, such as the various ways different doors can be unlocked (a_{31} or a_{32}).

At the lowest level, the leaf nodes of the T-AOG represent various relations, from poses of agents to locations of objects and agents.

The T-AOG can be learned in an unsupervised way from video [?].

A parse graph, pg, is an instance of the AOG with selections made at the Or-nodes, and provides a high-level interpretation of observed events. Parse graphs from the AOG of Figure **??** are shown over time in Figure **??**. In the T-AOG, fluent changes and causing actions happen independently.

SMOOTH OUT: DITCHED EQUATIONS, ASSUME PROBABILITY IS OUTPUT BY SEQUENCE OF ACTION/CHANGE DETECTIONS. ASSUME INDEPENDENT DISTRIBUTED. – THE LIKELIHOOD. CITE PAPERS FOR DETECTION. SIMPLIFY 2.1 A LOT.

The probability distribution of spatio-temporal parse graphs is a joint probability over the nodes given by and z_0 is the normalizing constant. In $\mathcal{E}_{\mathrm{T}}$, V^{OR_T} is the set of non-empty Or-nodes in pg_{T}, T is the set of terminal nodes in pg_{T}, and R is the set of temporal relations in pg_{T}. T consists of atomic actions composed of spatio-temporal relations between objects as well as fluents. λ_v, λ_t, and λ_{ij} give potential functions over the respective nodes and relations. λ_v again gives switch probabilities. For detailed explanation on the T-AOG and the probability distribution, we refer the reader to [?] and [?]. This current work takes these pg_{T} as input.

2.2 Representing causality locally: the C-AOG

EXPLAIN MORE. WHAT IS AND, WHAT IS OR. LEARNED (CITE MY PAPER). EXPLAIN HOW MANY PARAMETERS ARE USED IN C-AOG (NOT THAT MANY), HOW ARE THEY DECIDED. PUT TABLE: HOW MANY CAUSAL EVENTS WE HAVE.

CALL THIS ”INSTANTANEOUS”, NOT LOCAL

Identifying agent actions as causes for fluent changes as studied in cognitive science [?] and decomposing actions into fluents as used in vision to detect actions [?], the Causal And-Or Graph (C-AOG) provides a stochastic grammar representation of causality [?]. This graph integrates with And-Or representations used in vision to represent spatial and temporal knowledge [?], [?], [?].

The C-AOG represents a value for a fluent (e.g., $F(t) = \mathrm{ON}$ in Figure 4(a) where F is the monitor's display state) as a consequence of a temporally local INUS condition, an *insufficient* but *necessary* condition within a set of conditions that is *unnecessary* but *sufficient* for the effect [?]. In the C-AOG, Or-nodes represent the alternative means of causation (e.g., a monitor, through the computer, can be turned on by someone using a mouse or a keyboard). And-nodes group causing actions into single INUS conditions.

A parse graph (pg) in the C-AOG, such as that in Figure 4(b), is formed from a selection of the Or-nodes (such as the path shown by thicker, darker lines in (a)). It provides a causal interpretation for why fluent F has a particular value at time t, including fluent values and actions that lead to $F(t)$, as deemed relevant by the C-AOG..

Most frequently, the most probable causal parse graph for a given fluent at a given time is the inertial parse graph—the fluent maintained its value in the absence of a change-inducing action—such as shown in Figure 4(b) for the monitor power status.

EXPLAIN THIS EQUATION BETTER.

In [?], the C-AOG is learned in an unsupervised manner by linking actions from the T-AOG to fluents in the spatial domain. The probability model over the parse graphs available from the C-AOG builds on top of that of the T-AOG and is also defined through the

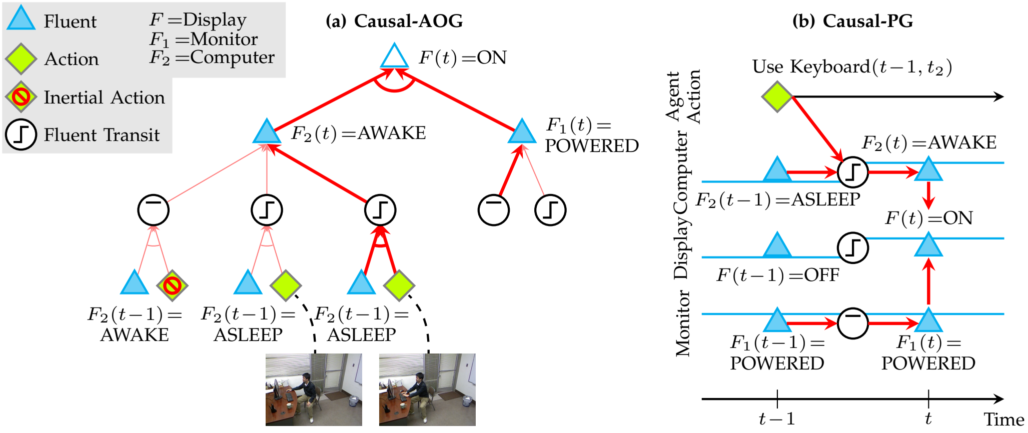


Fig. 4. (a) A C-AOG fragment for the display fluent to take the value ON. The value of the top level fluent at time t, $F(t)$, is a consequence of the values of its children. Children of And-nodes are connected by arcs. The fluent transit nodes indicate the kind of change that occurs in the fluent: step functions for change, flat lines for non-change. The inertial action, a lack of change inducing action, is shown by the non-action symbol. (b) In the parse graph, selections have been made on the Or-nodes of the C-AOG in (a). In this case, the display is determined to be on at t because someone used the keyboard, waking the computer. The monitor's power status (ON) does not change. TODO: separate this into two figures

energy. In particular, $p(pg_{\mathrm{C}}) = \frac{1}{Z}\exp\left(-\mathcal{E}\left(pg_{\mathrm{C}}\right)\right)$ where

$$\mathcal{E}(pg|I) = \sum_{t=1}^{n}\left(\mathcal{E}_{\mathrm{T}}(pg_t|I) + \sum_{v\in V_{\mathrm{C}}^{\mathrm{Or}}(pg_t)} \lambda_v(w(v))\right). \quad (1)$$

$\mathcal{E}_{\mathrm{T}}(pg_t|I)$ is the detection energy from event parsing for the included actions/fluents. $V_{\mathrm{C}}^{\mathrm{Or}}$ is the set of included Or-nodes in each local pg_t, $w(v)$ returns the selected branch, and λ_v gives the switch probability on the Or-nodes for the alternative causes as learned by maximum likelihood estimation. This Or-contribution to the energy specifies a prior on causality that favors the inertial action.

As we have now discussed two different parse graphs, we include the legend in Table 1 for referencing the parse graph notation over the next few sections.

TABLE 1
Legend of Parse Graph Notation

pg_{T}	temporal parse graph
pg	causal parse graph
$\mathbf{PG}$	sequence of causal parse graphs, $(pg_1, \ldots, pg_n)$
$pg_t(f_i, f_j)$	the causal parse graph (explanation) at time t where the fluent value at time $t-1$ was $F(t-1) = f_i$ and the fluent value at t is $F(t) = f_j$

3 Reasoning: The dynamic C-AOG

CALL OUT CONNECTION TO DYNAMIC BAYES NET (TRANSFER BETWEEN C-AOG, LIKE THEY TRANSITION BETWEEN BAYES NETS)

While the C-AOG as described above, referred to henceforth as a "local C-AOG fragment", represents the causal relations of a fluent at a single instant of time (using actions and conditions in a small window around that time), it lacks a mechanism to bind these instances forward and backward in time. Over the course of a sequence of events, however, knowing the fluent values at key points enables reasoning about the values of the fluents throughout the whole video sequence. Figure 5 summarizes this reasoning process. This section introduces the dynamic C-AOG for binding the temporally local C-AOG fragments together and for modeling the duration of the inertial action (when the fluent maintains status).

3.1, 3.2: EXPLAIN THE DURATION TERMS AND CONSISTENCY TERMS

]

3.1 Model formation and types of constraints

SET UP BACKGROUND FOR NEXT SECTION (INFERENCE).

TRANSITION PROBABILITY – HOW LIKELY TRANSITIONS BY ITSELF.

It is assumed that the observed fluents and actions in a given video sequence follow an underlying probability distribution, f, on $\mathbf{PG}$, the sequence $(pg_1, pg_2, \ldots, pg_n)$ of temporally local causal explanations at each timepoint of the video $i = 1, \ldots, n$.

While the distribution $f(\mathbf{PG})$ is unknown, it can be approximated with a model p by matching constraints to the observed data.

To initialize the process, the method takes observations from the distribution p_0 on local C-AOG fragments. In this paper, we fit two types of constraints: consistency terms, and duration histograms.

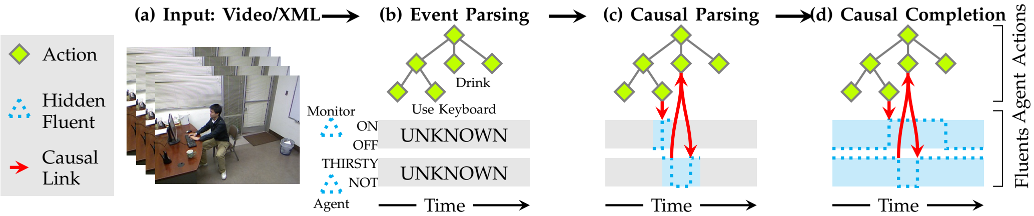


Fig. 5. Overview of the causal reasoning process using dynamic C-AOGs. For simplicity, only the monitor's display status and agent's thirst status are shown. (a) Event parsing results of a video are used as input. (b) Standard event parsing leaves many fluent values unassigned over the duration of the video. (c) Causal parsing assigns causal links between actions and fluents, allowing some fluent values to be filled in as preconditions or effects of agent actions. (d) Causal completion using the dynamic C-AOG fills in the fluents for the remaining times. TODO: change "causal completion" to "dynamic causal parsing"

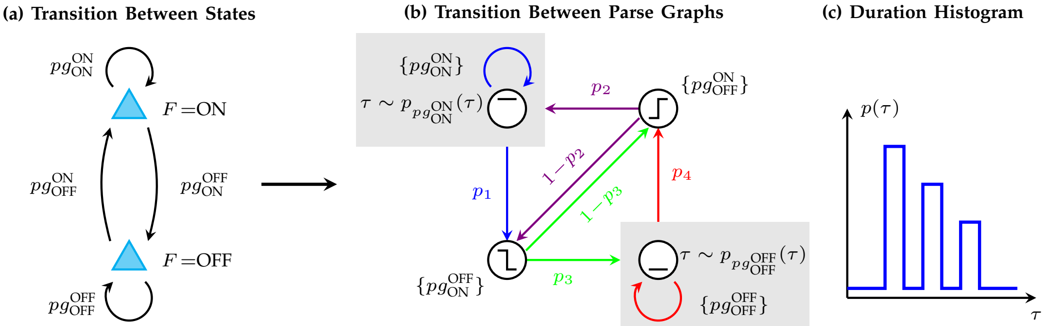

(a) Transition Between States
(b) Transition Between Parse Graphs
(c) Duration Histogram
$pg_{\text{ON}}^{\text{ON}}$
$F=$ON
$pg_{\text{OFF}}^{\text{ON}}$
$pg_{\text{ON}}^{\text{OFF}}$
$F=$OFF
$pg_{\text{OFF}}^{\text{OFF}}$
$\{pg_{\text{ON}}^{\text{ON}}\}$
$\tau \sim p_{pg_{\text{ON}}^{\text{ON}}}(\tau)$
p_2
$\{pg_{\text{OFF}}^{\text{ON}}\}$
p_1
$1-p_2$
$1-p_3$
p_4
$\{pg_{\text{ON}}^{\text{OFF}}\}$
p_3
$\tau \sim p_{pg_{\text{OFF}}^{\text{OFF}}}(\tau)$
$\{pg_{\text{OFF}}^{\text{OFF}}\}$
$p(\tau)$
τ

Fig. 6. (a) For a given set of values that a fluent can take (here, ON and OFF), the transitions between values also define the transitions between the parse graphs. (b) The dynamic C-AOG governs the transition between parse graphs by adding two features to the temporally local C-AOG fragments: a limitation on the available parse graphs and a duration capability for maintaining a given status. (c) Durations for maintaining status are sampled from histograms. In this histogram, the monitor is on and the most likely durations until a screensaver comes on are in even increments (such as multiples of 5 minutes). TODO: grey boxes... the set notation labels the node, the probability should label the arrow... switch! also, change notation to pg(ON, OFF), etc.

We first present the constraints used, and then we derive the dynamic C-AOG model, p, by matching the model to the data on the constraints.

3.2 Consistency constraints on local C-AOG fragments

CLARIFY!!!! MAYBE MOTIVATE THESE CONSTRAINTS WITH THE DIAGRAM FOR C-AOG.

As shown in Figure 6(a), a fluent transitions between values over time; the temporally local C-AOG fragments explain why each of these transitions occurs. At $t-1$, t, and $t+1$, the fluent F transitions between values $F(t-1)=f_i$, $F(t)=f_j$, and $F(t+1)=f_k$. At t and $t+1$, the transitions are causally described by local C-AOG fragments, $pg_t(f_i,f_j)$ and $pg_{t+1}(f_j,f_k)$ respectively. This transition between states also describes a transition between the local C-AOG parse-graph fragments, where the parse graph at $t+1$ is chosen from

$$\{pg(f_i,f_j)\} = \{pg : F(t)=f_j, F(t+1)=f_i\}, \tag{2}$$

which is a subset set of all the parse graphs for F.

earlier... describe what all parse graphs for F look like (in particular, those for ON, those for OFF, etc)

This transition gives rise to the first kind of constraint, the consistency constraint. For any two parse graphs, $pg(f_i,f_j)$ and $pg(f_k,f_l)$, let h_C calculate the consistency between the two as follows

$$h_C\left(pg(f_i,f_j), pg(f_k,f_l)\right) = \delta\left(f_j=f_k\right) = \begin{cases} 1, & \text{if } f_j=f_k \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

where $\delta\left(f_j=f_k\right)$ is the Dirac delta function.

In particular, the expected value under the model, E_p, for each transition between subsequent parse graphs, pg_t and pg_{t+1}, is therefore given by

$$E_p\left(h_C(pg_t, pg_{t+1})\right) = 1. \tag{4}$$

This constraint must hold for each $t=1,\ldots,n-1$, giving a total of $n-1$ consistency constraints.

The process of transition between the parse graphs is illustrated in Figure 6(b), where $F(t)$ has two possible values, ON and OFF, and each node is a temporally local parse-graph fragment from the C-AOG, such as shown in Figure 4(b). Following the consistency constraint, a parse graph of the type $pg(\text{ON}, \text{ON})$ can only transition to a parse graph of the form $pg(\text{ON}, \text{ON})$ or $pg(\text{ON}, \text{OFF})$.

The consistency constraints thus far described ensure a coherent solution to the reasoning process by limiting the way subsequent local parse graphs can be connected.

3.3 Duration constraints

To detect hidden fluents over long periods of time, the naive approach patches the temporally local C-AOG fragments together according to spatio-temporal detections, using only the consistency constraints thus far described. While this restricts the parse graphs available for time $t+1$, problems still arise. If action detections produce inconsistent fluent values, how to resolve the inconsistencies is unclear. In the naive approach, there is no way to specify whether it is more likely that a hidden change occurred between the two (and when that change occurred), or one of the detections was incorrect. For example, without knowledge of screensaver settings, it is unidentifiable whether the monitor display switches to screensaver automatically, after a specified period of time, or never. It is also necessary to represent an internal timer for the hidden fluents of the mind, such as an agent eventually developing thirst. Including a probabilistic timer mechanism to model the duration can fix these problems with the naive approach.

The standard solution of using a Markovian process to connect local parse graph fragments does not work in this case. A typical memoryless Markovian process, in which the transition probabilities are constant, leads to exponential fall-off for maintaining one state. Considering the monitor display fluent duration histogram shown in Figure 6(c), one can see how unreasonable the exponential fall-off is for fluents.

Therefore, transitions between the parse graphs representing different values for a fluent must be represented more generally than either the naive approach or a typical memoryless Markov process allow. Duration constraints are used to model how long a fluent can maintain a particular value, which is closely tied to the duration for which the inertial parse graph explains the fluent values. The process is illustrated in the grey boxes of Figure 6(b).

The duration constraints can take many forms, depending on the fluent. Below, we include a couple of examples of the constraints used for fluents in this paper, classified by their familiar resulting maximum entropy distributions. Let τ represent the duration a fluent, F, takes a given value, f.

Piecewise uniform. For fluents such as the screensaver, histograms were collected such as that shown in Figure 2(a). In this case, it was found that people set screensavers around five minute increments. After correcting for parse graphs with causing actions performed by agents, the duration a screensaver is off is modeled by the histogram shown. On each interval, I_i, with constant probability, the boolean for whether F repeatedly takes value f for a duration of τ within interval I_i is

$$h_{D,I_i}(\tau) = \delta(\tau \in I_i) \tag{5}$$

where i indexes the intervals. This gives a sequence of constraints where

$$E_p\left(h_{D,I_i}(\tau)\right) = p_i, \tag{6}$$

where p_i is the probability the duration falls in the respective section of the piecewise uniform distribution. For the screensaver, the constraints are given by the probabilities on the intervals, including ε around each five minute increment.

Uniform. For fluent values with no time dependence observable in the video (e.g., the light turning on/off), the uniform distribution is used to eliminate any importance of the duration. Even though the duration for an agent's thirst does not follow a uniform distribution, the uniform is justifiably used since the videos studied here are not long enough to warrant an expected pattern in agent's thirst.

The process of remaining in a local parse graph fragment is visualized in the grey boxes of Figure 6(b), where the duration is selected following a distribution $p(\tau)$.

The duration constraints described allow the consideration of seemingly inconsistent subsequent parse graphs by providing a mechanism for inserting a spontaneous change of the fluent value. We next combine the duration and consistency constraints with the distribution of the local C-AOG fragments in a single probability distribution.

4 Learning

DERIVE THE EQUATIONS/ENERGY (AND EXPLAIN PARAMETERS). CORRESPOND EACH TERM TO A STATISTICAL OBSERVATION FROM THE PREVIOUS SECTIONS (SECTION 3).

MAYBE: DITCH C-AOG FOR NOW (AND JUST GO WITH CONDITIONAL PROBABILITY... $\Delta F|A$.

INTRODUCE TRANSITION PROBABILITY. EITHER OBSERVATION OF STATUS CHANGE. SOME ARE INPUT, SOME ARE LEARNED THROUGH CAUSAL AOG LEARNING PROCESS.

POSS: IF ACTION HAPPENS, CHANGES TRANSITION PROBABILITY. IF THOSE 2 KIND OF TRANSITION PROBABILITIES CANNOT EXPLAIN THE FLUENT CHANGE (DURATION CANNOT EXPLAIN LIGHT ON/OFF), THE PROBABILITY TOO LOW/ZERO, THEN IT MUST INSERT THE ACTION THERE SO THAT TRANSITION HAPPENS. MAY SIMPLIFY INFERENCE SECTION... OVER TIME, FLUENT CHANGE BY ITSELF, FINE. THEY HAVE CERTAIN TRANSITION PROBABILITIES. IF THAT TRANSITION PROBABILITY IS REALLY REALLY LOW, OR IMPOSSIBLE, BUT IT STILL HAPPENS, THEN JUST SAY MUST INSERT ACTION THERE. SO HAVE **PRIOR PROBABILITY THERE TO PENALIZE INSERTING ACTIONS**. BUT GET REWARDED BY TRANSITION PROBABILITY. SO ALL I NEED IS TO SPECIFY THE STATUS PGI AND THE TRANSITION BETWEEN THEM. THEN THE WHOLE THING BECOMES DP, AND I'M DONE. OF COURSE, THE DP IS MOTIVATED BY ANOTHER PROCESS (DENY OR INSERT A NEW INPUT)

POSS: TRIVIALIZE CAOG INTO CONDITIONAL PROBABILITY, GIVEN ACTION OR NOT. THEN: CAN STILL HAVE C-AOG FOR FLUENT CHANGE BEFORE/AFTER. WHEN OBSERVE CERTAIN ACTION. THEN NEED TO KNOW THE FLUENT. DRINK CUP, INFER CUP HAD WATER. FOR THIS NEED C-AOG.

Each pg_t from a local C-AOG fragment causally explains the status for a single time t, and the sequence, $\mathbf{PG} = (pg_1, pg_2, \ldots pg_n)$, from the dynamic C-AOG explains the video.

Beginning with a distribution over the spatio-temporal parse graphs, p_{ST}, a model p is selected from Ω_p, the set of all models fitting the constraints listed above. Among all such p, the model closest to p_{ST} is selected, as measured by the KL-divergence. That is, p is selected such that

$$p = \text{argmin} KL(p||p_{\text{ST}}). \tag{7}$$

This method is the extension of the maximum entropy principle, where the reference distribution would be uniform. Equation 7 produces a model of the form

$$p(\mathbf{PG}|I) = \frac{1}{Z} p_{\text{ST}}(\mathbf{PG}|I) \exp\left(\sum_{\text{constraints}} \lambda_i h_i(\mathbf{PG}|I)\right). \tag{8}$$

where I represents the video (sequence of images), the λ_i are Lagrange multipliers, and Z is the normalizing constant. The constraints are used to solve for the λ_i.

This gives rise to an energy that is decomposed in terms of the durations, consistencies, and local C-AOG fragments:

$$\mathcal{E}(\mathbf{PG}|I) = \underbrace{\sum_{\tau} \mathcal{E}_D(\tau)}_{\text{duration}} + \underbrace{\sum_{t=2}^{n} \mathcal{E}_C(pg_{t-1}, pg_t)}_{\text{consistency}} + \sum_{t=1}^{n} \underbrace{\left(\mathcal{E}_{\text{T}}(pg_t|I) + \sum_{v \in V_C^{\text{Or}}(pg_t)} \lambda_v(w(v))\right)}_{\text{local C-AOG fragments}}, \tag{9}$$

Local C-AOG fragments. Given the durations and the transitions, the temporally local parse graph fragments are independent.

Duration. The duration terms enforce a timer for fluents that do not have causing actions by trying to add an event to explain a misperceived, or a spontaneous, change.

Consistency. Because a single local C-AOG fragment contains the fluent values at t and $t-1$, constraints are required to ensure coherence between subsequent local fragments. The consistency term is a hard constraint.

The probability, p, on $\mathbf{PG}$ causally describes the changes of the fluent jointly over time. Where the naive approach selects the value of a fluent based only on its value at the previous time, p allows reference to history beyond one step.

4.1 Visualizing the probability: The dynamic C-AOG

The graphical representation of the energy in Equation 9 is given by the dynamic C-AOG, as shown in Figure 6(b). The dynamic C-AOG incorporates the temporal-duration terms together with restrictions on the types of parse graphs available, controlling the selection of the local C-AOG parse graphs. Each node of the dynamic C-AOG is a temporally local causal parse graph. The dynamic C-AOG provides a graphical mechanism to visualize the transitions between these local parse graphs.

4.2 The joint causal/temporal AOG

The energy term in Equation 9 depends on the ST-energy provided. This ST-energy and parse graph pair provided through temporal parsing can consist of a single pair maximum or many pairs with a full distribution over ST-parse graphs. In the latter case, the energy allows joint selection of the optimal ST-parse graphs jointly with the causal explanations.

In the next section, we provide algorithms for inferring parse graphs in both situations.

5 Inference Algorithm

EXPLAIN WHERE TO INSERT, AND WHEN, AND WHICH ONE.

The optimal sequence, $\mathbf{PG}$, explaining the video is causally provided by

$$\mathbf{PG} = \text{argmax}_{\mathbf{PG}} P(\mathbf{PG}|\mathbf{I}). \tag{10}$$

Inference is performed by capitalizing on the long durations of inertial parse graph due to the infrequency of action and fluent information from event parsing. Starting with parse graphs and energies from event parsing, a set of potential causal explanations is first constructed by calculating energies for relevant temporally local C-AOGs. These potential fragments are then propagated over time, using an adaptation of the Viterbi algorithm. The compatible sequences of potential fragments are output, along with their energies.

The inference algorithm seeks to optimize the energy function in Equation 9 by adapting the traditional Viterbi algorithm to accommodate the non-Markovian duration terms. The temporally local causal parse graphs form the hidden state nodes that the traditional Viterbi algorithm would navigate over. Departing from the traditional Viterbi algorithm, the algorithm presented here includes a nested minimization within each step. This minimization attempts to insert a spontaneous change due to the duration term, if such is warranted, and is illustrated in Figure 7. We begin by first formalizing the adapted Viterbi algorithm, and then we provide methods for the joint inference of causal and temporal parse graphs.

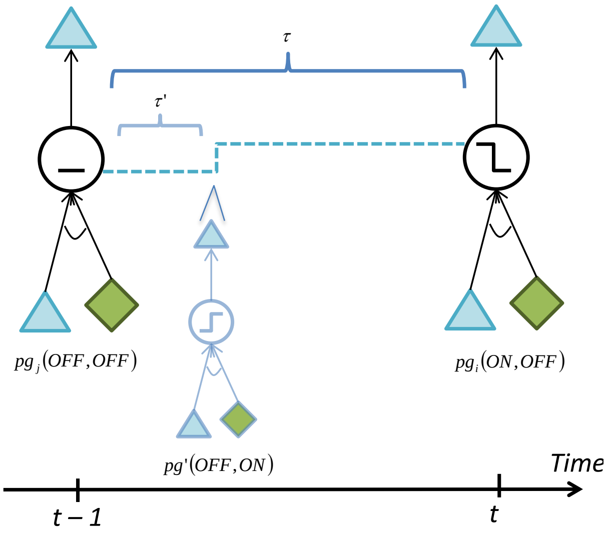


Fig. 7. Insertion step of inference algorithm, exampled on the screensaver fluent. The Viterbi algorithm tries to find optimal subsequent parse graphs based on the temporally local and consistency energy contributions. Before accepting the new energy of the parse graphs connecting the two important time points, a new parse graph is optimally inserted between them. If the insertion improves the energy, the insertion is kept. This process promotes chains that would otherwise be inconsistent where the duration constraint warrants, and can demote seemingly consistent chains that violate duration constraints. TODO: make pretty in tikz

5.1 Assumptions

We begin by only examining “important” time points, that is, time points where either a change in a fluent or a causing action is detected. This reduces the necessary search space. In general, all instances between these important time points are best explained by the causal parse graph with the inertial action: the fluent maintains status because no change-inducing action occurred. For simplicity, these so-called important time points are numbered consecutively, indexed with t.

The one exception to the generalization regarding the parse graphs between important time points occurs when a “spontaneous” change is instigated due to the duration term. That is, the duration term can be responsible for a change in fluent values for a given fluent, F, between those time points. We make the additional assumption that the duration term can be responsible for at most one change in fluent values for F between those time points. This assumption is reasonable for the fluents studied here, where a spontaneous change happens for only one of the fluent values in those cases. For example, there is a perceived spontaneous change to ON for the screensaver when no agent is using the computer, but only actions can turn the screensaver OFF. Similarly, an agent’s thirst appears to be turned ON spontaneously, but OFF is activated by drinking.

These two assumptions reduce computation time, and allow for a more efficient algorithm. By only examining important time points, we reduce the search space for the algorithm. Further, by limiting the number of spontaneous fluent changes, we are able to adapt the Viterbi algorithm for the non-Markovian process, as we develop in the next section.

5.2 Adaptation of the Viterbi algorithm

Our adaptation of the Viterbi algorithm finds the most likely sequence of local C-AOG fragments that results in the given sequence of input ST-parse graphs, shown in Figure 8.

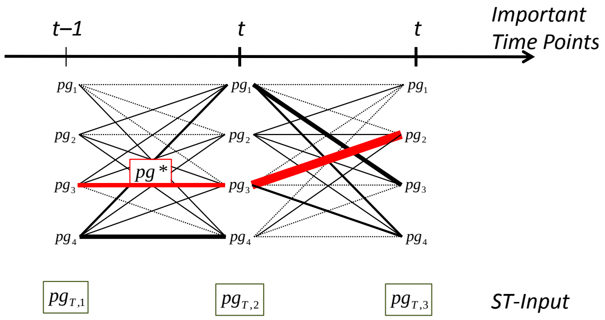


Fig. 8. The adapted Viterbi algorithm. The most probable path is computed recursively. Here, the path includes a local C-AOG fragment inserted because of the energy contribution of the duration terms. TODO: make pretty in tikz

Given the local parse graph fragment at important time point $t-1$ and the duration the fluent has maintained state due to the inertial action, the energy of the most probable path to the next important time point, δ_t, is recursively given by

$$\delta_t(pg_j, \tau) = \min_{pg_i} \left(\delta_{t-1}(pg_j) + \mathcal{E}(I|pg_j) + \mathcal{E}^*(pg_i, pg_j, \tau) \right), \tag{11}$$

where $\mathcal{E}(I|pg_j)$ is the observation energy described in 5.2.1 and $\mathcal{E}^*(pg_i, pg_j, \tau)$ is the energy from duration and consistency constraints described in 5.2.2. The recursion is initialized with the energies from the temporally local fragments at the first important time point.

5.2.1 Energy contribution from the local C-pg

The third term, $\mathcal{E}(I|pg_j)$, is the energy resulting from parsing on the temporally local C-AOG fragments. The local C-AOG contribution to the energy of Equation 9 consists of a ST-energy from the ST-parsing process and a prior energy. These energies add to form the local C-AOG energy.

When a fluent change or an action is detected (i.e., at an important time point), the energies are examined for all potentially relevant local C-AOG fragments. Local fragments are deemed relevant if (a) they contain the action or fluent in question as one of their nodes, or (b) they have the same top-level fluent as those already included.

If the fluent change is detected before the action, it is assumed that the causing action must have already occurred or that the fluent change was mis-detected. Both cases thus require the consideration of all local parse graph fragments with the same top-level fluent. In this case, all relevant local parse graphs are completed: energies are tallied (using a prior when there is no detection information), the fluent change is considered irrelevant to the future, and the relevant local parse graphs are considered as the hidden state nodes for the time point.

If the action is detected first, however, new important time points are continually considered until the distance in time exceeds a pre-set latent time. The latent time is set based on the fluent, and "instantaneous" fluents have the shortest latent times. If the fluent change is detected within the latent time, all parse graphs for the top-level fluent are completed as explained above. If the fluent change is not detected before the latent time runs out, then the local parse graphs are completed at the end of the latent time, and the action is considered dealt with.

These completed parse graphs form the states for each important time point that is tracked throughout the adapted Viterbi algorithm. Parsing on the local C-AOG fragments provides competing pg to causally explain each fluent change and/or action at the important time point.

5.2.2 Energy from duration and consistency constraints

The nested minimization, $\mathcal{E}^*$, incorporates the consistency terms, and attempts to insert a spontaneous event. $\mathcal{E}^*$ compares the effect on the energy of such an insertion between two local parse graphs at consecutive important time points, $pg_i(a,b)$ and $pg_j(c,d)$.

The proposed insertion considers the duration that the fluent has maintained the value b, weighing possible inconsistencies against the length of duration. We let τ denote the amount of time the sequence has thus far maintained fluent value b. In particular, if $a \neq b$, then $pg(a,b)$ represents a change in fluent, and $\tau = 0$. However, if $a = b$, then $\tau > 0$.

[[EQUATION 14 (NOW 12) UNCLEAR. BBEFORE INSERTION/AFTER INSERTION–DOING THE CHANGE LIKE A MONTECARLO JUMP. BUT THIS EQUATION DOESN'T SOUND LIKE A JUMP. BEFORE INSERT, HAVE STATUS A; AFTER INSERT, GO TO STATUS B. STATUS A AND B HAVE DIFFERENT ENERGIES. IN MONTE CARLO SITUATION, WE JUST ACCEPT WITH PROBABILITY. BECAUSE IN MOST COMPLICATED CASE, MIGHT WANT TO DO GIBBS SAMPLER OR METROPOLIS JUMP.

Under an insertion, the lowest possible energy depends on the consistency contributions, as well as duration contributions before and after the insertion:

$$\begin{aligned}\mathcal{E}_I^* = \min_{\substack{\tau' < \tau \\ pg'}} \Bigg(& \underbrace{\mathcal{E}_C(pg_i(a,b), pg') + \mathcal{E}_C(pg', pg_j(c,d))}_{\text{Consistency}} \\ & + \underbrace{\mathcal{E}_D(\tau') + \tau' \mathcal{E}(pg_{\text{Inert}}(b,b))}_{\text{Before Insertion}} \\ & + \underbrace{\mathcal{E}_D(\tau - \tau') + (\tau - \tau')\mathcal{E}(pg_{\text{Inert}}(c,c))}_{\text{After Insertion}} \Bigg).\end{aligned} \tag{12}$$

where $\mathcal{E}_C$ and $\mathcal{E}_D$ are the energies from Equation 9. $pg_{\text{Inert}}(b,b)$ is the non-action inertial parse graph maintaining fluent value b. τ is tracked from the end of the last non-action parse graph, and is only relevant at the start of the next non-inertial parse graph.

Without an insertion, on the other hand,

$$\mathcal{E}_{NI}^* = \mathcal{E}_C(pg_i, pg_j) + \mathcal{E}_D(\tau) + \tau \mathcal{E}(pg_{\text{Inert}}(b,b)). \tag{13}$$

$\mathcal{E}^*$ is then computed

$$\mathcal{E}^* = \min\left(\mathcal{E}_I^*, \mathcal{E}_{NI}^*\right) \tag{14}$$

When there is no significant duration term, this reduces to

$$\mathcal{E}^* = \mathcal{E}_{NI}^*. \tag{15}$$

If $\mathcal{E}^* = \mathcal{E}_I^*$, then the best temporal location to insert a parse graph, along with the best inserted parse graph is given by:

$$\phi_t^*(pg', \tau') = \text{argmin}_{pg', \tau'} \left(\mathcal{E}^*(pg_i, pg_j, \tau) \right). \tag{16}$$

Further, the corresponding most probable parse graph at t given by

$$\phi_t(pg_j) = \text{argmin}_{pg_i} \left(\delta_{t-1}(pg_j) + \mathcal{E}^*(pg_i, pg_j, \tau) \right). \tag{17}$$

The preceding equations give rise to the inference algorithm for the dynamic C-AOG. This algorithm is summarized in Algorithm 1.

Algorithm 1: Dynamic causal inference

```
Input  : ST-pg from video, and probabilities
Output : Most probable sequence of C-pg
foreach important time point $t$ do
    Complete all temporally local causal parse graphs;
    if $t > 2$ then
        Calculate $\mathcal{E}_I^*$, $\mathcal{E}_{NI}^*$, and $\mathcal{E}^*$ according to
        Equations 12, 13, and 14;
        Create hashes by Equations 17 and 16;
    end
end
Return lowest energy (most probable) $\mathbf{PG}$;
```

5.3 Joint inference

Joint temporal and causal inference is performed through a beam search over the temporal parse graphs. The top performing temporal parse graphs are used, one at a time, as input for the dynamic causal parsing process of Algorithm 1. This added step is summarized in Algorithm 2.

Algorithm 2: Joint temporal and causal inference

```
Input  : Sequence of top $n$ ST-pg from video, and
         probabilities
Output : Jointly most probable sequence of T-pg and C-pg
foreach ST-pg do
    Compute most probable sequence of C-pg by Alg. 1;
end
Return the overall most probable $\mathbf{PG}$;
```

6 Consequences of inference

While the main output of the inference is to find the jointly most probable temporal and causal explanation of the observed events and fluents, the inference process described can be used in deeper ways.

6.1 Special fluents: preconditions, trigger conditions, and relative fluents

Using the dynamic C-AOG, causal explanations are used to reason hidden fluent values throughout the video. This section describes how the capacity of the dynamic C-AOG is expanded by adding three new types of fluents to the local C-AOG fragments: preconditions, trigger conditions, and relative fluents.

The local C-AOG fragments can be extended to include key fluent preconditions for a richer description of the causing state. The original C-AOG only considered agent actions as INUS causing conditions. However, by examining observations with and without preconditions, preconditions can be learned and added to the INUS conditions. This allows, for example, the monitor's display state to be represented in terms of its power status and the computer's power status, as in Figure 4(a).

One important type of precondition is the internal fluent triggering the agent to take action. Including fluents of the mind as preconditions in the C-AOG fragments allows them to be inferred.

With hidden measurable fluents, such as the amount of water in a cup, it is almost impossible to determine an exact amount, and for daily cognition tasks, it is usually not even desirable to do so. Reasoning can be accomplished by considering certain actions that are known to increase or decrease the quantity, e.g., dispensing water into or drinking water from a cup. Therefore, measurable quantities are described in terms of a new concept, the relative fluent. The relative fluent compares the value of the fluent at t to that of $t-1$ in making a determination for time t. If $F(t) > F(t-1)$, the relative fluent value at time t is MORE. If $F(t) < F(t-1)$, the value is LESS, and if $F(t) = F(t-1)$, the value is SAME. Adding local C-AOG fragments for these relative fluents allows relative values of measurable fluents to be reasoned over time.

6.2 Joint inference: inferring hidden fluents and action

In addition to expanding the types of fluents considered in the local C-AOG fragments, we can also deepen the effects of the inference process by using the calculated $\mathbf{PG}$ to infer values of hidden, or missing, fluents and actions.

7 Experiments

[[MDS – DO 2-3 FLUENTS AT A TIME.]]

[[MAYBE: SIDE-BY-SIDE BAR CHARTS FOR DISTRIBUTION COMPUTER COMES UP WITH FOR PEOPLE; FOR DIFFERENT FLUENTS?]]

[[BASELINE: ACTION ALONE VS ME. OR FLUENT ALONE VS. ME]]

[[ANOTHER COMPARISON: GROUNDTRUTH COLLECTION–ANNOTATION PROCESS]]

[[ARGUE BECAUSE CAUSAL AND 3D, THIS IS INVARIANT/TRASNPORTABLE]]

[[GET PR CURVES BY COMPARING HOW CLOSE WE ARE TO 'NEAREST' HUMAN JUDGEMENT.]]

[[CHOP INTO DIFFERENT SECTIONS – REORGANIZE. PUT A TABLE OF WHAT THE EVENT/ACTIONS WE CONSIDER.]]

[[IDEA FOR COMPARISON: COMPARE TO HMMS]]

[[GO INTO: WE AVOID EARLY DECISIONS]]

[[CLAIM IN PAPER: WE'RE GOING TO RELEASE THE DATASET]]

[[FIRST SENTENCE: WE HAVE A NEW DATASET FOR YOU GUYS. BASED ON HOW I CHOPPED THE EVENTS, INCLUDING 5 SCENES. EACH ROOM INVOLVES SO MANY ACTIONS. HOW MANY FUNCTIONAL OBJECTS, HOW MANY STATUSES (FUNCTIONAL OBJECTS), AND HOW MANY CLIPS OF THE DATA I HAVE.]]

[[LIST ALL 4 CASES (HAVE F,A; HAVE F, NO A; HAVE A, NO F; HAVE NEITHER). HOW DID I DO ON EACH? – MENTION LIST IN THE BEGINNING – THE DIFFERENT KINDS OF DETECTIONS WE CAN DO. NEED TABLE HERE FOR HOW STC IMPROVE EACH OTHER.]]

[[HOW MANY DATA ARE USED TO TRAIN. DESIGNED THE CAUSAL-AOG BY HAND.]]

[[POSE AND CAUSAL TRAINED ON OTHER? FLUENT BY CLASSIFIER (DISCRIMINITIVE TRAINING]]

[[HOW WAS DETECTIONS DONE?]]

[[ASK PING/BRUCE: WHAT WAS USED FOR TRAINING DATA? ONLY MY TRAINING DATA, OR DID YOU INCORPORATE OTHER CLIPS? – TRANSPORTABLE ONLY IF THEIR TRAINING CONTAINED OTHER DATA]]

[[BASELINE: SIMPLIFY COMPONENTS OF MODEL, AND SHOW PERFORMANCE AGAINST HUMANS—WHAT IF MEMORYLESS MARKOV PROCESS WAS USED INSTEAD?]]

7.1 The data

TODO: discuss how T parsing done... poses are clustered, given semantic labels, fed into the t-parsing program mingtian made off mingtao's work. beam search.

TODO: if possible, run the office data through mingtian's code. i now understand that ping was just doing low-level detections, and perhaps mingtian's event parsing can smooth out the jumpy results (upon which mine were completely dependent because there was no fluent detections)

In order to evaluate the methods presented for reasoning values of hidden fluents, video data was captured using a Kinect in two scenes: a hallway and an office. The combined video totals approximately 20 minutes. A summary of the fluents contained in the video, as well as the values each fluent can take, is included in Figure 2. Using video spanning a long duration allows demonstration of the reasoning process. The values each fluent can take are separated into a discrete set with granularity based upon the types of queries to be answered, e.g., Cup Fluent ∈ {MORE, LESS, SAME}.

TABLE 2
List of fluents, separating hidden fluents (top) from observable (bottom).

Office	Hallway
Trashcan: MORE/LESS/SAME Monitor Display: ON/OFF Monitor Power: ON/OFF Computer: ASLEEP/AWAKE Phone: ACTIVE/STANDBY Cup: MORE/LESS/SAME Water Stream: ON/OFF Agent : THIRSTY/SATIATED Agent: HAS_TRASH/NOT	Trashcan: MORE/LESS/SAME Water Stream: ON/OFF Phone: ACTIVE/STANDBY Agent: HAS_PHONE/NOT Agent: THIRSTY/SATIATED Agent: HAS_TRASH/NOT
Light: ON/OFF	Light: ON/OFF

Using grammar models to parse the video, a spatio-temporal (ST) description of objects and events was generated and output to the XML files included in the supplemental materials. These ST descriptions and corresponding probabilities were used as input. Inconsistencies typical of vision detections are represented in the XML. In particular, in the hallway dataset, a light is detected as changing ON/OFF, but no action is detected.

The ST description for the hallway dataset contains observable fluent detections as well as actions. The office data only contains detections of actions (no fluent detections), and, because of this, observable fluents are treated as hidden.

7.2 Misinformation: correcting spatio-temporal detections

In the hallway dataset, multiple changes in the light fluent were detected, yet no causing action was detected and logged in the XML, presenting a common situation in vision—detections are usually imperfect. The method described in this paper corrects these errors by balancing the maintenance of detections with the consistency of causal explanations. Figure 9 shows typical candidates of the results sorted in order of probability. Because it is unknown which detections in the dataset are incorrect, it is important to maintain multiple possible interpretations from causal parsing.

7.3 Reference Estimates

To evaluate performance on hidden and unobserved fluents, we compare the predicted fluent values by our algorithm with two types of reference estimates.

Human Estimates. A human experiment is performed through a website, allowing participants to complete the task at their own leisure. Each of 15 participating subjects is shown the test video which pause at preset frames, e.g., those shown in Figure 10. At each key frame, the subject is asked to assign a total of 100 points to all possible values of each fluent, according to his/her own (human) recognition and reasoning for events/actions and fluent changes. Assignment of the points correspond to the subjective probabilities of the fluent values, e.g., 90 points are assigned to COMPUTER ASLEEP if the subject feels it is 90% likely that the computer is ASLEEP. The subject is allowed to revise previous judgments with information derived from subsequent frames. The recorded human judgments are taken as ground truth with a degree of variance, with which we can examine the performance of our algorithm against the distribution of human data.

Baseline Estimate. For a baseline estimate, the hidden fluents were assigned without using any detection or causal information. Their values are uniformly assigned, e.g., 50% for LIGHT ON and 50% for OFF. The baseline estimate is completely conservative without inclination to any observation or prior belief. The baseline estimate provides a discriminative reference against which we

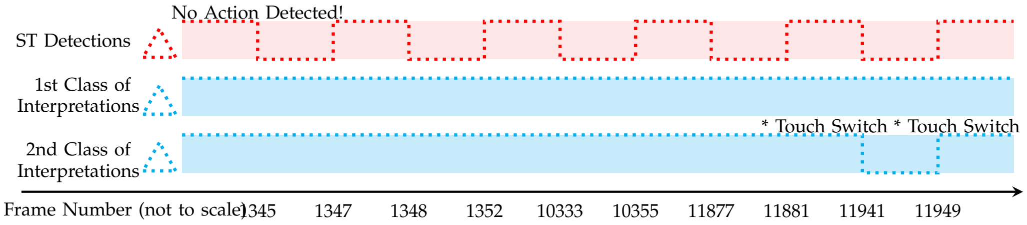


Fig. 9. Given an ST detection that moves the light fluent between ON and OFF without a causing action, the reasoning process prefers this to be explained by incorrect detections of the light fluent. The second most probable explanation is that two of the changes had causing actions that were missed by the detection.

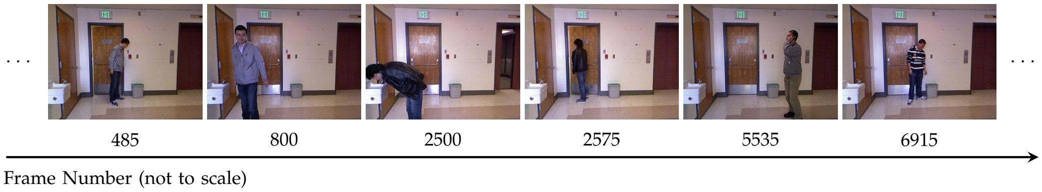


Fig. 10. Sample of human judgment key frames.

can see how well our algorithm approximates human performance. This reference point is very informative given the limited amount of human data (too limited to derive a reliable confidence interval with) whose true distribution is unknown.

TODO: use/include better baselines

7.4 Experimental Results: inferring hidden fluents forward and backward in time

Table 3 shows numerically the performance of our algorithm in comparison to human and baseline. Values in the table are summarized based on $d(a,b) = \sum_{ij} d_{ij}(a,b)$, an accumulated distance between two estimates a and b over each fluent i at key frame j, where d_{ij} is the total variation (TV) distance between two assignments of the 100 points (equivalent to discrete probability distributions).

TABLE 3
Comparison of computer, human and baseline estimates by TV distances. c, b and h represent computer, baseline and human estimates, respectively. s and t are indices for the collection of human estimates.

	Hallway	Office
computer-baseline, $d(c,b)$	67.00000	32.50000
computer-human, mean$_s$ $d(c,h_s)$	37.50067	58.67119
human-baseline, mean$_s$ $d(h_s,b)$	57.41622	66.39667
human-human, mean$_{s\neq t}$ $d(h_s,h_t)$	33.46714	31.26088

TODO: include measures of how close the computer was to the human it was closest to.

TODO: maybe include distance computer was to median/mean human result

For better visualization, these estimates (computer, human, and baseline) are reduced to two dimensions using multi-dimensional scaling (MDS) according to the TV distance between estimates, and plotted in Figure 11.

The hallway dataset contains observable fluent detections as well as actions. Both contribute to the causal inference of hidden fluents. The computer performance is very similar to human performance according to both the data and the plot. The baseline is away from the cluster of both. The office dataset only contains detections of actions and all fluents are hidden and can only be inferred using the actions. The result is less satisfactory than that of the hallway data, but the computer performance still improves over the baseline towards human performance, as shown in Figure 11(b).

7.5 Non-Markovian nature of data

TODO: show screensaver example, and show how markov process fails.

7.6 Joint parsing: causal and temoral information

TODO: incorporate with the misinformation section/link more strongly, but point out difference. corrections vs filling in missing information.

TODO: show a few of instances: 1) light in hallway where we have temporal parsing... using causal parsing improves results to ones in-line with humans. showcases C to T 2) light in office. no spatial detection of light, so causal parsing completely dependent on (poor) detection of action. cautions about poor T. 3) a completely indetectable fluent, such as cup level. so with the caveat that detection of hidden fluents is potentially unreliable without good action detection, it is possible to infer the hidden.

8 Analysis of Experiments

TODO: explain that the humans were far from each other, allowing leniency for the computer as well. For

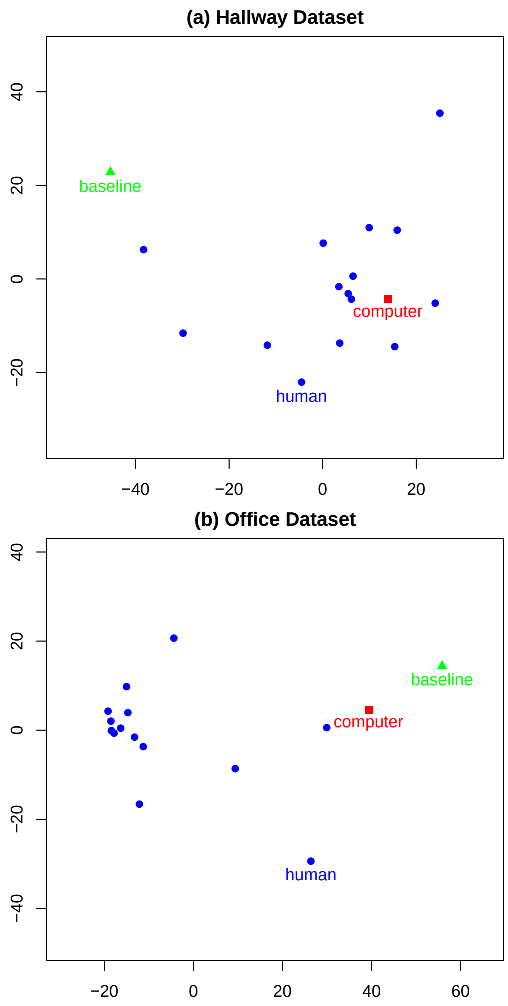
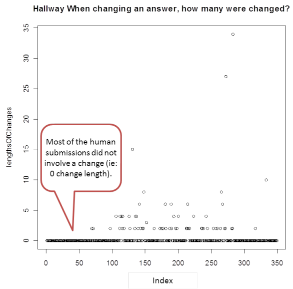
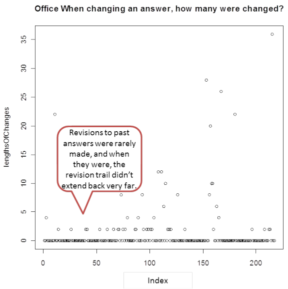


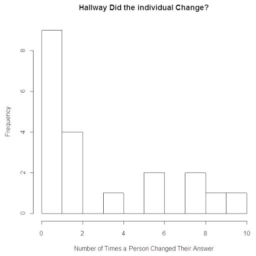
Fig. 11. X-Y scatter plots of the 2D-embedded fluent value estimates using MDS. Red squares: estimates using our algorithm. Blue dots: human estimates. Green triangles: baseline estimates.

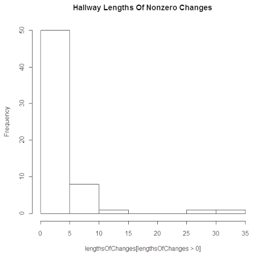
the MDS, humans were actually quite far from each other for some of the check points, which gave the computer leeway overall. Even with this disparity, the baseline confirms our performance is more in line with human estimations.

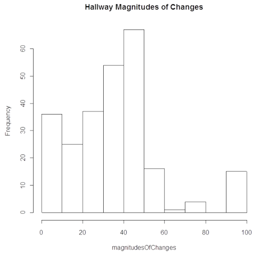
TODO: pick and choose graphs to include

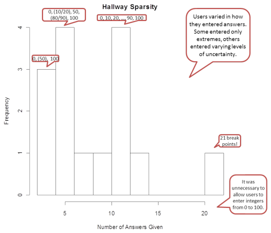


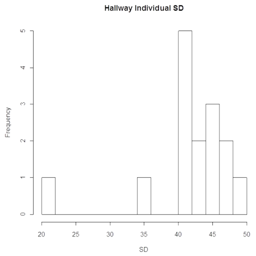


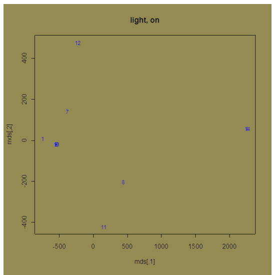


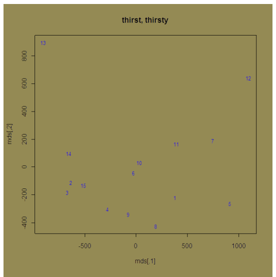


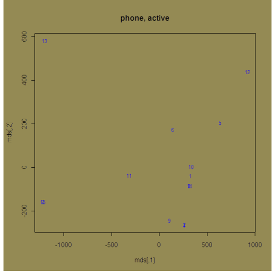


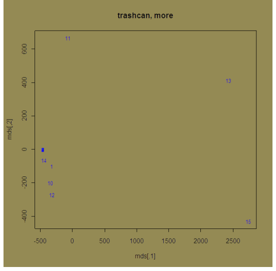












9 Discussion and Summary of Contributions

This paper has introduced the dynamic C-AOG, which can infer the values of hidden fluents in video over a long duration, combining interpretations of multiple events backward and forward in time and binding these detections into a coherent sequence of explanations.

In addition to reasoning about top-level fluent consequences of actions, fluents were considered in terms of preconditions for an action to have an effect, in terms of internal trigger conditions (such as an agent's state of mind), and in terms of relative fluents where values were scalar rather than binary.

In experiments, inference was evaluated against baseline and human judgments. The input to the system consisted of a single (incorrect by human judgment) ST description of the events. For the hallway dataset, the system was presented with inconsistent information that it was able to reason through and find an explanation of the scene comparable to human judgments. For the case of the office dataset, event detections were poor and no fluent detections were available to identify conflicts, leaving the system heavily dependent on those incorrect event detections. Despite this disadvantage, the system was still able to use reasoning to outperform the baseline.

The results for the office dataset highlight the consequences of assuming too much from a single interpretation and reinforce the need to have multiple interpretations. This paper presents some foundations for implementing the inference of hidden fluents. Using this framework, joint inference of events and fluents can be used to create a feedback system, allowing better detection of events.

ACKNOWLEDGMENTS

The authors would like to thank... TODO. Don't forget Ping's work (low-level action detection with the Kinect).

PLACE PHOTO HERE

Amy Morrow received a BA degree in Mathematics from UC Berkeley in 2002, and an MS degree in Mathematics from SFSU in 2004. After a brief stint as a community college instructor, she is currently a PhD student in Statistics at UCLA, working in the VCLA lab. Her research interests include statistical models for vision and causality.

PLACE PHOTO HERE

Mingtian Zhao Biography text here.

PLACE PHOTO HERE

Song-Chun Zhu received a BS degree from the Univ. Sci. & Tech. of China in 1991 and a PhD degree from Harvard University in 1996. He is a professor with the Department of Statistics and the Department of Computer Science at UCLA. His research interests include computer vision and learning, statistical modeling, and stochastic computing. He received a number of honors, including the David Marr Prize in 2003 with Z. Tu et al., the J.K. Aggarwal prize from the Intl Association of Pattern Recognition in 2008, the Marr Prize honorary nominations in 1999 and 2007 with Y.N. Wu et al., a Sloan Fellowship in Computer Science in 2001, a US National Science Foundation Early Career Development Award in 2001, and an US Office of Naval Research Young Investigator Award in 2001. In 2005, he founded, with friends, the Lotus Hill Institute for Computer Vision and Information Science in China as a nonprofit research organization (www.lotushill.org). He is a Fellow of IEEE.