Temporal Monitoring of Agent Beliefs Under Uncertainty with Subjective LTL

Benjamin Herd[1], **Jessica Kelly**[1], **Reinhard Stolle**[1]

[1]Fraunhofer Institute for Cognitive Systems (IKS), Munich, Germany
{benjamin.herd,jessica.kelly,reinhard.stolle}@iks.fraunhofer.de

Abstract

ML-based agents acting in open environments must form and revise beliefs under pervasive perceptual and epistemic uncertainty. Verifying such agents over time therefore requires reasoning about how their uncertain beliefs evolve. However, existing temporal verification and monitoring frameworks – including probabilistic and multi-valued logics – typically reason about Boolean or graded truth of propositions along system runs, rather than about an agent's internal belief state. We propose Subjective LTL (SLTL), a temporal specification language whose atomic propositions are Subjective Logic opinions, i.e., tuples of belief, disbelief, and uncertainty about predicates in an agent's symbolic knowledge model. Under an evidential semantics, temporal operators aggregate evidence over time about persistent hypotheses, so SLTL formulas constrain how the agent's belief state should evolve. This supports temporal verification and monitoring directly over the dynamics of knowledge-grounded, uncertainty-aware agents, and can equally be used to equip such agents with introspective monitors over their own evolving beliefs, which we illustrate on a simple example.

Introduction

Knowledge-grounded semantic agents such as robotic assistants or autonomous vehicles operate in open, dynamic environments and use ML-based functions (e.g., perception) that provide noisy and partial information. At the same time, they are often grounded in symbolic models such as ontologies or knowledge graphs. Trustworthy behaviour in such hybrid systems requires reasoning not only about what is true in the environment, but about what the *agent believes* to be true, and how these beliefs evolve over time under uncertainty.

Runtime verification and temporal monitoring based on Linear Temporal Logic (LTL) provide a principled way to specify and check system behaviour over execution traces. However, temporal specifications are typically interpreted over traces of propositions that are treated as Boolean (or thresholded) facts at each time step. This is only partially sufficient for agents whose world model is constructed from uncertain perception: properties such as "whenever the agent believes there is an obstacle, it eventually believes it is stopping" are themselves about *belief states*, not crisp events. Classical LTL can only operate on ad hoc Boolean abstractions of these beliefs which ignores uncertainty.

This paper argues that temporal verification for knowledge-grounded semantic agents should be carried out *directly over structured belief states*, rather than over Boolean abstractions. We propose *Subjective LTL (SLTL)*, a temporal specification language in which atomic propositions are interpreted as *binomial opinions* in Subjective logic (SL). These opinions encode belief, disbelief, and uncertainty about binary predicates in the agent's symbolic model (e.g., `pedestrian(x)` or `obstacle-at(x,y)`). SLTL thus specifies and monitors properties over the evolution of the agent's epistemic state, instead of over raw observations or thresholded classifier outputs. Beyond external verification, this also enables agents to monitor and reflect on their own evolving belief states, using SLTL properties as introspective specifications.

However, replacing Boolean propositions with SL opinions has a significant impact on the semantics of the specification language. There are now a range of possible interpretations, one of which – the *evidential interpretation* – we focus on in this paper. Instead of treating each time-indexed occurrence of a proposition as an independent uncertain event, we view predicates as *persistent hypotheses* (e.g., "pedestrian detected"). Temporal operators are defined via SL belief fusion and thus combine opinions from successive observations about such hypotheses. In this view, an SLTL formula constrains not just *when* a property should hold, but *how the agent's confidence in it is allowed to evolve* as new evidence arrives. Interpreting temporal properties in this evidential way enables *temporal verification of belief trajectories*. A monitor evaluates SLTL formulas over traces of evidence-producing states, each contributing new opinions about predicates in the agent's knowledge model. Resulting temporal opinions can be turned into runtime verdicts via tunable thresholds on belief, disbelief, or derived probabilities, supporting explainable trade-offs between caution and permissiveness in safety-critical settings. We make the following contributions:

- We introduce *Subjective LTL (SLTL)*, a temporal specification language whose atoms are binomial SL opinions over predicates in an agent's symbolic knowledge model, enabling temporal reasoning over uncertain beliefs.
- We develop an *evidential interpretation* of SLTL, in

which temporal operators are defined in terms of evidence fusion over time for persistent hypotheses. Temporal evidence aggregation is *parametric* in the chosen SL fusion operator, so the meaning of temporal operators can be tailored to different monitoring objectives; we deliberately do not fix a single default.

- We outline a progression-based *runtime monitoring algorithm* for SLTL that uses short-circuiting and threshold-based decision rules over opinions to derive temporal verdicts on evolving belief trajectories.

Background

This section briefly recalls Linear Temporal Logic (LTL) and its finite-trace variant, which we use as the temporal backbone of SLTL, and Subjective Logic (SL), which provides the underlying representation of uncertain beliefs.

LTL and Finite-Trace Semantics

Temporal logics for system specification are commonly divided into branching-time logics (such as CTL, CTL*) and linear-time logics such as LTL (Baier and Katoen 2008). Branching-time logics model time as a tree of possible futures, whereas LTL views time as a single (typically infinite) linear sequence of states. We consider here LTL(F,G), a strictly less expressive subset of LTL which excludes the *until* operator. We adopt this fragment because it is sufficient for the safety and liveness patterns we target in this paper; extending SLTL with an explicit *until* operator is left for future work. The syntax of this fragment is:

$$\phi ::= \top \mid p \mid \neg\phi \mid \phi \,\square\, \phi \mid \mathbf{X}\phi \mid \mathbf{F}\phi \mid \mathbf{G}\phi \tag{1}$$

where $\square \in \{\wedge, \vee\}$ and p ranges over a finite set of atomic propositions, $\wedge, \vee, \neg$ are the Boolean connectives, $\mathbf{X}$ is the *next* operator, $\mathbf{F}$ is the *eventually* operator, and $\mathbf{G}$ is the *always* operator.

LTL formulae are interpreted over infinite paths $\pi = s_0 s_1 s_2 \ldots$ of states. Intuitively, at state s_i, $\mathbf{X}\phi$ holds if ϕ holds at s_{i+1}, $\mathbf{F}\phi$ holds if ϕ holds at this or some future state and $\mathbf{G}\phi$ holds if ϕ holds at this and all future states. The temporal operators admit the well-known *expansion laws* shown in Table 1, which decompose a formula into obligations on the current and the next state.

Runtime verification and the finite-trace problem In *runtime verification* (Leucker and Schallhart 2009), temporal properties are checked against concrete *execution traces* produced by a running system or simulator. In contrast to model checking, which explores all possible executions, runtime verification focuses on the single observed run and reports satisfaction or violation as soon as possible.

$$\mathbf{X}\phi = \top \wedge \mathbf{X}\phi \tag{2}$$
$$\mathbf{F}\phi = \phi \vee \mathbf{XF}\phi \tag{3}$$
$$\mathbf{G}\phi = \phi \wedge \mathbf{XG}\phi \tag{4}$$

Table 1: Expansion laws for LTL(F,G).

Due to this focus on individual traces, time is viewed as a linear flow, and properties are typically written in LTL or a close variant. Monitors often exploit the expansion laws in Table 1 to operate in a *forward-oriented* manner: in each step they identify an *immediate obligation* that must hold in the current state and a *future obligation* that must hold in the remainder of the trace. As soon as the immediate obligation is satisfied and no future obligation is generated, the whole formula is satisfied; similarly for refutation.

A practical complication arises because traces produced by testing or simulation are typically *finite*, whereas classical LTL is defined over infinite paths. Under the standard semantics of $\mathbf{X}$, the formula $\mathbf{X}\phi$ at the last state of a finite trace is neither satisfied nor refuted, because there is no successor state on which to evaluate ϕ. This ambiguity propagates to $\mathbf{F}$ and $\mathbf{G}$ and makes it impossible, e.g., to conclusively refute a formula $\mathbf{F}\phi$ or to conclusively satisfy $\mathbf{G}\phi$ on a finite trace.

Finite LTL (FLTL) FLTL (Manna and Pnueli 2012) addresses the complications of evaluating temporal operators on finite traces by refining the $\mathbf{X}$ operator into a *universal* and an *existential* variant. On non-final states of a trace, the two coincide; at the final state, they differ:

- $\mathbf{X}_\forall\phi$ (universal next) is evaluated as *vacuously true* if there is no successor state.
- $\mathbf{X}_\exists\phi$ (existential next) is evaluated as *false* if there is no successor state.

The syntax of FLTL(F,G) is:

$$\phi ::= \top \mid p \mid \neg\phi \mid \phi \,\square\, \phi \mid \mathbf{X}_\forall\phi \mid \mathbf{X}_\exists\phi \mid \mathbf{F}\phi \mid \mathbf{G}\phi \tag{5}$$

where $\square \in \{\wedge, \vee\}$. The corresponding expansion laws are shown in Table 2. We use this fragment as the temporal basis for SLTL.

Subjective Logic

Subjective Logic (SL) (Jøsang 2016) is a framework for reasoning under uncertainty that combines ideas from probability theory and Dempster–Shafer evidence theory. Its core data structures are *subjective opinions* representing an agent's belief, disbelief, and uncertainty about the truth of a proposition. SL provides algebraic operators for combining and transforming opinions, which we use to lift temporal operators to uncertain beliefs. In this paper, we restrict attention to *binomial* opinions over binary domains $\mathbb{X} = \{x, \bar{x}\}$.

Definition 1 (Binomial opinion)**.** *Let $\mathbb{X} = \{x, \bar{x}\}$ be a binary domain. A* binomial opinion *about x is a tuple $\omega_x = (b_x, d_x, u_x, a_x)$ where:*

- b_x (belief) *is the belief mass supporting x being* true*;*
- d_x (disbelief) *is the belief mass supporting x being* false*;*
- u_x (uncertainty) *is the remaining, uncommitted belief mass and the complement of* confidence *$(1 - u_x)$;*
- a_x (base rate) *is the a priori probability of x in the absence of committed belief.*

where $b_x, d_x, u_x, a_x \in [0, 1]$ and $b_x + d_x + u_x = 1$[1].

[1]Although u_x is uniquely determined by b_x and d_x via normalization, we follow standard SL practice in treating (b_x, d_x, u_x) as a primitive triple, which makes uncertainty explicit in the semantics.

$$\mathbf{X}_\forall \phi = \top \wedge \mathbf{X}_\forall \phi \text{ in non-final states, else } \top \quad (6)$$
$$\mathbf{X}_\exists \phi = \top \wedge \mathbf{X}_\exists \phi \text{ in non-final states, else } \bot \quad (7)$$
$$\mathbf{F}\phi = \phi \vee \mathbf{X}_\exists \mathbf{F}\phi \quad (8)$$
$$\mathbf{G}\phi = \phi \wedge \mathbf{X}_\forall \mathbf{G}\phi \quad (9)$$

Table 2: Expansion laws for FLTL(F,G).

We have that $\omega_\top = (1, 0, 0, a)$ and $\omega_\bot = (0, 1, 0, a)$ denote absolute opinions with full belief/disbelief, respectively, and $\omega_V = (0, 0, 1, a)$ denotes a vacuous opinion with full uncertainty.

Constructing opinions from evidence SL provides a simple way to construct binomial opinions from counts of supporting and contradicting evidence. Let r be the number of positive observations supporting x, s the number of negative observations supporting $\bar{x}$, and $W > 0$ a *non-informative prior weight*[2]. The resulting opinion $\omega_x = (b_x, d_x, u_x, a_x)$ is given by:

$$b_x = r/(r + s + W), \quad (10)$$
$$d_x = s/(r + s + W), \quad (11)$$
$$u_x = W/(r + s + W). \quad (12)$$

Thus, uncertainty u_x shrinks as evidence accumulates, while the ratio between b_x and d_x reflects the balance of positive and negative observations.

Binomial opinions have a probabilistic interpretation in terms of Beta distributions. Let $p(x)$ denote the (unknown) probability that x is true. Then ω_x corresponds to a Beta density $\mathrm{Beta}(p(x); \alpha, \beta)$ with parameters $\alpha = r + aW$ and $\beta = s + (1 - a)W$ whose expectation is

$$E[x] = \frac{\alpha}{\alpha + \beta} = \frac{r + aW}{r + s + W} = b_x + a \cdot u_x.$$

Due to this link, SL opinions can be viewed as structured summaries of the underlying evidential uncertainty.

Combining opinions SL defines a family of operators for combining and transforming binomial opinions (Jøsang 2016). In this work we rely on:

- *Complement* $\neg\omega_x$, which corresponds to negation of the underlying proposition.
- *Multiplication* $\omega_x \cdot \omega_y$, which represents the SL equivalent of *logical conjunction (AND)* of opinions about *different* domains.
- *Co-multiplication* $\omega_x \sqcup \omega_y$, which represents the SL equivalent of *logical disjunction (OR)* of opinions about *different* domains.
- *Belief fusion* $\omega_x^A \circledast \omega_x^B$, which represents an aggregation of opinions formed by evidence about the *same* domain collected by observers A and B. SL provides a range of fusion operators, e.g. cumulative fusion ($\oplus$), averaging fusion ($\underline{\oplus}$), constraint fusion ($\odot$), or consensus & compromise fusion (ⓒ).

In our approach, the complement, multiplication, and co-multiplication operators play the role of Boolean $\neg$, $\wedge$, and $\vee$ operators at the level of opinions; the fusion operators are central to the *evidential interpretation* of SLTL, where temporal operators like $\mathbf{F}$ and $\mathbf{G}$ will be treated as repeated fusion of evidence over time about persistent hypotheses.

[2] W controls how quickly uncertainty decreases as evidence accumulates. For binary domains it is often set to $W = 2$, corresponding to adding one virtual positive and one virtual negative observation.

SLTL: Reasoning About Uncertain Beliefs

We now introduce *Subjective LTL (SLTL)*, an extension of FLTL in which atomic propositions range over binomial opinions rather than Boolean truth values. SLTL provides a temporal specification language for *belief trajectories*: it constrains how an agent's uncertain beliefs about symbolic predicates may evolve along a finite trace.

Extending LTL with SL allows us to distinguish between (i) uncertainty about the truth of events at a given time and (ii) uncertainty arising from incomplete or unreliable evidence accumulated over time. This leads to two semantic readings. Under the *logical interpretation*, opinions are treated as uncertain truth values of time-indexed events and temporal operators are lifted truth-functionally, closely mirroring FLTL. In this paper, we focus on the *evidential interpretation*, where repeated observations of the same formula are treated as evidence for or against a persistent hypothesis (such as "there is an obstacle"), and temporal operators accumulate this evidence over time using SL fusion.

Syntax

Let AP be a finite set of atomic predicates over the agent's symbolic state. SLTL formulae are defined by[3]:

$$\varphi \ ::= \top \mid p \mid \neg\varphi \mid \varphi \,\square\, \varphi \mid \mathbf{X}_\forall \varphi \mid \mathbf{X}_\exists \varphi \mid \mathbf{F}\varphi \mid \mathbf{G}\varphi \quad (13)$$

where $\square \in \{\wedge, \vee\}$ and $p \in AP$. The temporal operators $\mathbf{X}_\forall$ and $\mathbf{X}_\exists$ are the weak and strong versions of $\mathbf{X}$, and $\mathbf{F}$ and $\mathbf{G}$ are treated as primitive.

Evidential Semantics

A finite trace $\tau = (\sigma_0, \sigma_1, \ldots, \sigma_n)$ is an *epistemic trajectory*, i.e. a sequence of *epistemic states* about the world. Each state $\sigma_i : AP \to Op$ maps atomic predicates (denoting facts about the world) to binomial opinions, i.e. $\sigma_i(p) = (b_p, d_p, u_p, a_p)$ (denoting an agent's belief about p). Intuitively, $\sigma_i(p)$ is the agent's opinion (e.g., based on perception) at time i about whether p holds.

We write $[\![\varphi]\!]_i \in Op$ for the *evidential valuation* of φ at position i of τ. This is interpreted as an opinion about a *persistent hypothesis* H_φ (e.g. "φ holds eventually/always") given the evidence up to and beyond i. The semantics uses SL complement $\neg\omega$ for logical negation, SL multiplication $\omega_1 \cdot \omega_2$ for conjunction, SL co-multiplication $\omega_1 \sqcup \omega_2$ for disjunction, and SL belief fusion $\omega_1 \circledast \omega_2$ where

[3] Similar to FLTL, we focus here on the (F,G) fragment.

$\circledast \in [\oplus, \underline{\oplus}, \odot, ⓒ, \cdots]$ is one of the SL belief fusion operators used to aggregate evidence over time. The flexibility with respect to belief fusion is a deliberate design choice in SLTL. It is treated as a semantic parameter that is chosen by the user to reflect the intended monitoring objective and evidential assumptions (e.g., whether observations are independent, how quickly evidence should accumulate, how to handle conflicts). Different instantiations of $\circledast$ thus give rise to different, but related, readings of the same temporal formula: they change *what question* the formula asks about the belief trajectory rather than merely how it is computed. We deliberately do not prescribe a single default operator; the choice of $\circledast$ is part of the design of an SLTL specification. The evidential valuation for the non-temporal and the 'next' operators are defined recursively as follows:

$$[\![\top]\!]_i = \omega_\top \tag{14}$$

$$[\![p]\!]_i = \sigma_i(p) \tag{15}$$

$$[\![\neg\varphi]\!]_i = \neg\, [\![\varphi]\!]_i \tag{16}$$

$$[\![\varphi_1 \wedge \varphi_2]\!]_i = [\![\varphi_1]\!]_i \cdot [\![\varphi_2]\!]_i \tag{17}$$

$$[\![\varphi_1 \vee \varphi_2]\!]_i = [\![\varphi_1]\!]_i \sqcup [\![\varphi_2]\!]_i \tag{18}$$

$$[\![\mathbf{X}_\forall\varphi]\!]_i = \begin{cases} [\![\varphi]\!]_{i+1} & \text{if } i < n, \\ \omega_V & \text{if } i = n, \end{cases} \tag{19}$$

$$[\![\mathbf{X}_\exists\varphi]\!]_i = \begin{cases} [\![\varphi]\!]_{i+1} & \text{if } i < n, \\ \omega_\perp & \text{if } i = n, \end{cases} \tag{20}$$

Negation, conjunction, and disjunction are interpreted as truth-functional lifts of the corresponding Boolean operators to opinions, using SL complement for $\neg$, multiplication for $\wedge$, and co-multiplication for $\vee$. The $\mathbf{X}$ operators follow the FLTL semantics: at final position n, the expansions of $\mathbf{X}_\exists$ and $\mathbf{X}_\forall$ collapse to $\omega_\perp$ and ω_V, respectively. For $\mathbf{X}_\exists$, this yields a violation of the property and matches the FLTL semantics. For $\mathbf{X}_\forall$, this produces a maximally uncertain verdict that can be interpreted as "no further information is available beyond the end of the trace".

The remaining temporal operators ($\mathbf{F}$ and $\mathbf{G}$), in contrast, have an *evidential reading*: they are evaluated as opinions about persistent hypotheses that the agent forms about the world. Concretely, they can be understood as follows:

- $\mathbf{F}\varphi$: does collected evidence about φ *eventually support* the statement that φ holds (= existential reading)?
- $\mathbf{G}\varphi$: does collected evidence about φ *continuously support* the statement that φ holds (= universal reading)?

To represent the collection of evidence, we first introduce an accumulation function that uses belief fusion as follows.

$$\mathcal{A}_i^F(\varphi) = \begin{cases} [\![\varphi]\!]_i & \text{if } i = 0, \\ \mathcal{A}_{i-1}^F(\varphi) \circledast [\![\varphi]\!]_i & \text{if } i > 0, \end{cases} \tag{21}$$

$$\mathcal{A}_i^G(\varphi) = \begin{cases} [\![\varphi]\!]_i & \text{if } i = 0, \\ \mathcal{A}_{i-1}^G(\varphi) \circledast [\![\varphi]\!]_i & \text{if } i > 0, \end{cases} \tag{22}$$

The temporal operators $\mathbf{F}$ and $\mathbf{G}$ can now be defined in terms of the accumulation function as follows.

$$[\![\mathbf{F}\varphi]\!]_i = \begin{cases} [\![\varphi]\!]_i & \text{if } \mathrm{sat}(\mathcal{A}_i^F(\varphi)), \\ [\![\varphi]\!]_i \circledast [\![\mathbf{X}_\exists\mathbf{F}\varphi]\!]_i & \text{else} \end{cases} \tag{23}$$

$$[\![\mathbf{G}\varphi]\!]_i = \begin{cases} [\![\varphi]\!]_i & \text{if } \mathrm{ref}(\mathcal{A}_i^G(\varphi)), \\ [\![\varphi]\!]_i \circledast [\![\mathbf{X}_\forall\mathbf{G}\varphi]\!]_i & \text{else} \end{cases} \tag{24}$$

Both $\mathbf{F}\varphi$ and $\mathbf{G}\varphi$ are evaluated over the same epistemic agent trajectory τ; what differs is the persistent hypothesis (existential vs. universal) and possibly the fusion operator used to aggregate the opinions from τ into a temporal opinion. This aggregation is controlled by the fusion operator $\circledast$ in Eqs. (23) and (24). Different instantiations of $\circledast$ correspond to different *evidential profiles* of the temporal operators, i.e., to different questions about the belief trajectory:

- If $\circledast$ is instantiated with a fusion operator that *accumulates* independent pieces of evidence (e.g. cumulative fusion), the resulting opinions characterise the agent's *ability to learn* from repeated opportunities: can it eventually gather enough evidence to support or refute "φ happens at least once" (for $\mathbf{F}\varphi$) or "φ keeps holding" (for $\mathbf{G}\varphi$)?
- If $\circledast$ is instantiated with a fusion operator that *harmonises* or *stabilises* dependent evidence (e.g. averaging or consensus & compromise fusion), the resulting opinions emphasise the *stability* of the belief trajectory: does the belief about φ remain consistent over time, despite fluctuations in local observations? For $\mathbf{F}\varphi$, such a choice answers questions like "does the agent maintain stable, non-oscillating confidence that φ will eventually hold, rather than exhibiting brief spikes of belief?". For $\mathbf{G}\varphi$, it addresses whether "the agent's belief that φ currently holds stays consistently high over time, without frequent drops that would indicate an unstable invariant".

The evidential semantics of $\mathbf{F}$ and $\mathbf{G}$ is therefore *parametric* in the choice of $\circledast$ at specification time; this adds modelling flexibility and allows SLTL specifications to be tuned to concrete monitoring objectives and domain assumptions.

The functions $\mathrm{sat}(\cdot)$ and $\mathrm{ref}(\cdot)$ referred to in Eqs. (23)–(24) implement the (F)LTL-style short-circuit behaviour: for $\mathbf{F}\varphi$, once $\mathrm{sat}([\![\varphi]\!]_i)$ returns true at some i, we set $[\![\mathbf{F}\varphi]\!]_i = [\![\varphi]\!]_i$ and stop fusing with future states, so later negative evidence cannot dilute the fact that "eventually φ" has already been satisfied; conversely, for $\mathbf{G}\varphi$, once $\mathrm{ref}([\![\varphi]\!]_i)$ returns true, we set $[\![\mathbf{G}\varphi]\!]_i = [\![\varphi]\!]_i$ and stop fusing, so subsequent positive evidence cannot revert a decisive violation of "always φ". This aligns the evidential semantics with the intuitive existential versus universal readings of $\mathbf{F}$ and $\mathbf{G}$. Possible choices for $\mathrm{sat}(\cdot)$ and $\mathrm{ref}(\cdot)$ are discussed below.

Setting Satisfaction and Refutation Thresholds

In the presence of SL-based binomial opinions with explicit belief, disbelief, and uncertainty masses, defining a criterion for satisfaction or refutation is non-trivial and needs to be decided based on the chosen application scenario. Functions $\mathrm{sat}(\cdot)$ and $\mathrm{ref}(\cdot)$ act as decision rules that turn opinions into short-circuit verdicts. They can be instantiated in different ways, depending on the desired attitude to risk and uncertainty (e.g., conservative vs. permissive). We sketch a few typical but neither exhaustive nor normative patterns below.

Belief/disbelief thresholding. A simple choice is to trigger satisfaction when belief is high and refutation when disbelief is high, optionally under a bound on residual uncertainty:

$$\mathrm{sat}(\omega_x) \equiv (b_x \geq \theta_b \wedge u_x \leq \theta_u)$$
$$\mathrm{ref}(\omega_x) \equiv (d_x \geq \theta_d \wedge u_x \leq \theta'_u)$$

for application-specific thresholds $\theta_b, \theta_d, \theta_u, \theta'_u \in [0,1]$. This matches the intuition that "eventually" succeeds once there is sufficiently strong evidence for φ, while "always" fails once there is sufficiently strong evidence against φ.

Margin- and uncertainty-aware decisions. To avoid decisive verdicts when belief and disbelief are similar, margins can be defined as follows:

$$\mathrm{sat}(\omega_x) \equiv (b_x - d_x \geq \Delta_{\text{sat}} \wedge u_x \leq \theta_u)$$
$$\mathrm{ref}(\omega_x) \equiv (d_x - b_x \geq \Delta_{\text{ref}} \wedge u_x \leq \theta'_u)$$

with $\Delta_{\text{sat}}, \Delta_{\text{ref}} > 0$. High uncertainty u then blocks both satisfaction and refutation and prevents premature short-circuiting.

Expectation-based thresholds. Using the expectation value of an opinion, one may instead define

$$\mathrm{sat}(\omega_x) \equiv (E[x] \geq \theta_p)$$
$$\mathrm{ref}(\omega_x) \equiv (E[x] \leq \theta'_p)$$

where $E[x] = b_x + a \cdot u_x$ is the expected probability of predicate x being true. In practice, different temporal operators may use different decision rules or thresholds. In the examples further below we instantiate $\mathrm{sat}(\cdot)$ and $\mathrm{ref}(\cdot)$ with a simple belief/disbelief-threshold rule with fixed parameters for illustration; designing richer, domain-specific decision rules is left to future work.

Runtime Monitoring

The evidential semantics defined in the previous section can be operationalised in a progression-based fashion suitable for runtime monitoring. Conceptually, the expansion laws decompose each temporal obligation into (i) an *immediate obligation* that is checked in the current state, and (ii) a *residual obligation* that is passed to the next step. In non-final states, this residual obligation propagates forward; at the end of the trace, $\mathbf{X}_\exists$ and $\mathbf{X}_\forall$ inject the boundary opinions $\omega_\perp$ and ω_V, respectively, so that the final evidential verdicts for $\mathbf{F}\varphi$ and $\mathbf{G}\varphi$ are well-defined on finite traces. For runtime monitoring, we use a similar approach that, at each step, computes (i) an opinion over the current prefix and (ii) a residual temporal obligation to be checked on the remaining suffix. Listings 1 and 2 sketch this progression-based monitor in a schematic way, omitting data-structure details (e.g., how opinions for subformulae are stored across steps). They illustrate the progression pattern (immediate vs. residual obligations, boundary handling, and short-circuiting) rather than a fully fleshed-out implementation.

For $\mathbf{X}_\forall$ and $\mathbf{X}_\exists$, the offline semantics in Eqs. (19)–(20) expresses that the opinion at position i is whatever opinion φ will receive at position $i+1$. A strictly progression-based online monitor cannot know this at position i, so in non-final states it (i) propagates φ as a residual obligation and (ii) returns a vacuous immediate opinion ω_V for the 'next' formula (see Listing 1, lines 16–21). At the final state, boundary opinions ω_V and $\omega_\perp$ are injected.

Listing 1: Function EVAL performing an evidential evaluation of an SLTL formula at time $t = i$

```
1  Input: formula ψ, epistemic state σ_i, flag final,
       fused 'finally' opinion acc_F, fused 'globally'
       opinion acc_G
2  Output: Immediate verdict opinion ω, residual
       obligation ψ'
3
4  if ψ is atomic p then ω ← σ_i(p); ψ' ← ⊤
5  else if ψ is ¬φ then ω, ψ' ← ¬EVAL(φ);
6  else if ψ is φ1 ∧ φ2 then
7    ω1, ψ'1 ← EVAL(φ1)
8    ω2, ψ'2 ← EVAL(φ2)
9    ω ← ω1 · ω2 // apply SL multiplication
10   ψ' ← ψ'1 ∧ ψ'2
11 else if ψ is φ1 ∨ φ2 then
12   ω1, ψ'1 ← EVAL(φ1)
13   ω2, ψ'2 ← EVAL(φ2)
14   ω ← ω1 ⊔ ω2 // apply SL co-multiplication
15   ψ' ← ψ'1 ∨ ψ'2
16 else if ψ is X∀φ then
17   if final then ω ← ω_V; ψ' ← ⊤
18   else ω ← ω_V; ψ' ← φ
19 else if ψ is X∃φ then
20   if final then ω ← ω_⊥; ψ' ← ⊤
21   else ω ← ω_V; ψ' ← φ
22 else if ψ is Fφ then
23   if sat(acc_F) then ω ← ω_⊤; ψ' ← ⊤
24   else ω ← ω_V; ψ' ← X∃Fφ
25 else if ψ is Gφ then
26   if ref(acc_G) then ω ← ω_⊥; ψ' ← ⊤
27   else ω ← ω_V; ψ' ← X∀Gφ
28 return (ω, ψ')
```

In order to produce a verdict as soon as possible in a true online fashion, the monitor shown in Listing 2 implements the short-circuit behaviour encoded by $\mathrm{sat}(\cdot)$ and $\mathrm{ref}(\cdot)$. For $\mathbf{F}\varphi$, as soon as $\mathrm{sat}(\cdot)$ holds on the accumulated opinion acc_F at some position i, verdict $\omega_\top$ is returned and the monitor stops fusing with later states. Dually, for $\mathbf{G}\varphi$, once $\mathrm{ref}(\cdot)$ holds, verdict $\omega_\perp$ is returned and the monitor stops, since a decisive violation of "always φ" has been observed. The resulting opinion is then mapped to three-valued verdicts (*satisfied*, *refuted*, *inconclusive*).

Note that the evaluation of SLTL formulas under the evidential semantics on finite traces is decidable: for any finite trace τ and formula φ, the temporal opinion $[\![\varphi]\!]_i$ is effectively computable, because all SL operators are total functions given by fixed arithmetic operations on finite opinion tuples. Our progression-based runtime monitor runs in time $\mathcal{O}(n \cdot |\varphi|)$ and space $\mathcal{O}(|\varphi|)$ on a trace of length n and a formula of size $|\varphi|$ (number of subformulas), assuming constant-time SL operations.

Listing 2: A progression-based runtime monitor for SLTL

```
Input: property φ, growing trace (σ_0, σ_1, ...)
Output: after each σ_i: output opinion ω_i and verdict
     v_i ∈ {"satisfied","refuted","inconclusive"}

ψ ← φ; v ← "inconclusive";
acc_F ← σ_0; acc_G ← σ_0;
for each new state σ_i do
    final ← ( σ_i is last state of trace )
    (ω_i, ψ) ← EVAL(ψ, σ_i, final, acc_F, acc_G)
    acc_F ← acc_F ⊛ ω_i
    acc_G ← acc_G ⊛ ω_i
    if ω_i is ω_⊤ then v ← "satisfied"; break
    else if ω_i is ω_⊥ then v ← "refuted"; break
    else v ← "inconclusive"
return (ω_i, v)
```

Example Evaluation

In this section, we illustrate the behavior of SLTL and visualize how different evidence fusion strategies influence the satisfaction of the temporal properties $\mathbf{F}\varphi$ and $\mathbf{G}\varphi$. To this end, we construct representative discrete-time opinion traces of length T, where each time step corresponds to the arrival of new evidence. The traces are designed to exhibit characteristic patterns (e.g., stable confidence, sudden degradation, and recovery) to highlight the effect of temporal operators and fusion dynamics. Temporal operators are evaluated over finite traces using the bounded semantics described above. Note that the scenarios are intentionally stylised and designed to cover characteristic patterns of belief evolution rather than to model a particular real-world system.

Scenario and Trace

For this example, we consider an agent that represents a decision making component in a safety-critical system, e.g. an autonomous vehicle. The agent's opinions encode its subjective belief about environment- and safety-related predicates and evolve over time as new evidence becomes available. To investigate how temporal properties interact with qualitatively different opinion dynamics, we consider the evolution of the agent's opinions under two distinct input scenarios, each with a different opinion trace behaviour.

Across both scenarios, opinion traces are initialized with prior weight $W = 2$ and base rate $a = 0.5$ with a finite trace length of $|\tau| = 80$. We define satisfaction and refutation thresholds for the temporal operators $\mathbf{F}\varphi$ and $\mathbf{G}\varphi$ using belief-based thresholding, fixing the belief threshold to $\theta_b = 0.8$ and defining satisfaction and refutation as:

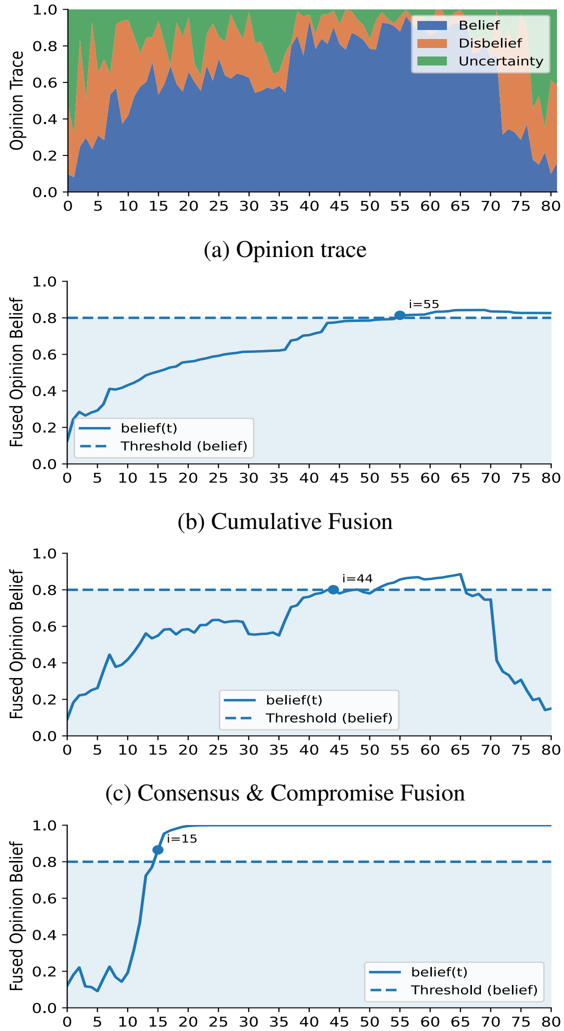
$$\begin{aligned} \mathrm{sat}(\omega_x) &\equiv (b_x \geq \theta_b), \\ \mathrm{ref}(\omega_x) &\equiv (b_x < \theta_b). \end{aligned}$$

For illustrative purposes, refutation is defined with respect to the belief threshold rather than disbelief, allowing clearer visualization of threshold crossings in the opinion traces. We instantiate $\circledast$ using three SL-based fusion strategies: *constraint fusion*, modeling conservative evidence conjunction; *cumulative fusion*, modeling evidence accumulation over time; and *consensus & compromise fusion*, balancing agreement and conflict. These strategies induce distinct temporal dynamics in belief, disbelief, and uncertainty. To ensure reproducibility, opinion traces are generated using a fixed random seed. Experiments were run using Python 3.10.12 with NumPy 2.2.3.

Results

Scenario 1: Accumulating evidence for an eventual event In Scenario 1, the agent holds an opinion about a predicate obs: "there is an obstacle in front of the vehicle". The agent's confidence in the presence of an obstacle gradually increases as supporting evidence accumulates, before decreasing again as contradicting evidence becomes available. This results in a fused opinion trajectory where, regardless of the choice of fusion operator, the opinion temporarily exceeds a given threshold but does not remain consistently above it. A visualization of this opinion trace can be seen in Fig. 1a. The corresponding fused opinion trajectory for Scenario 1 under *cumulative fusion* is shown in Fig. 1b. As supporting evidence is incrementally aggregated over time, the agent's confidence in the predicate obs increases gradually, causing the belief to cross the specified threshold only at a relatively later timestep, $i = 55$. In contrast, Fig. 1c shows the fused opinion trajectory obtained using *consensus & compromise fusion*. By explicitly balancing agreement and conflict between incoming evidence, this strategy amplifies consistent support more quickly, resulting in an earlier threshold crossing compared to cumulative fusion. Finally, Fig. 1d depicts the opinion evolution under *constraint fusion*. Constraint fusion rapidly concentrates belief when evidence is mutually reinforcing, leading to the earliest threshold crossing among the three strategies. These results highlight how fusion strategy choice directly influences temporal satisfaction of $\mathbf{F}\varphi$.

Scenario 2: Accumulating evidence for persistent safety In this scenario, the agent holds an opinion about the predicate $safe$: "the current situation is safe". As shown in Fig. 2a, the opinion is initially confident and stable over an extended period, reflecting consistent evidential support. This confidence is then disrupted by a sudden drop caused, for example, by a brief misclassification in a perception module, after which the opinion gradually recovers as further supporting evidence is accumulated. Figs 2b–2d show the corresponding evolution of the fused belief for the temporal hypothesis under three SL fusion operators: cumulative fusion, constraint fusion, and consensus & compromise fusion. In each case, the solid blue curve depicts the fused belief over fusion steps, and the dashed line marks the belief threshold $\theta_b = 0.8$ used by ref to short-circuit $\mathbf{G}\varphi$. For cumulative and constraint fusion, the fused belief trace remains above the threshold at all steps, so the monitor never triggers $\mathrm{ref}(\cdot)$ and $\mathbf{G}\varphi$ is satisfied: occasional conflicting observations are treated as admissible disturbances that are outweighed by the strong overall evidence for safety. Under consensus & compromise fusion, the same underlying conflict causes a deeper transient drop in the fused belief, pushing it below the threshold and thereby refuting $\mathbf{G}\varphi$.



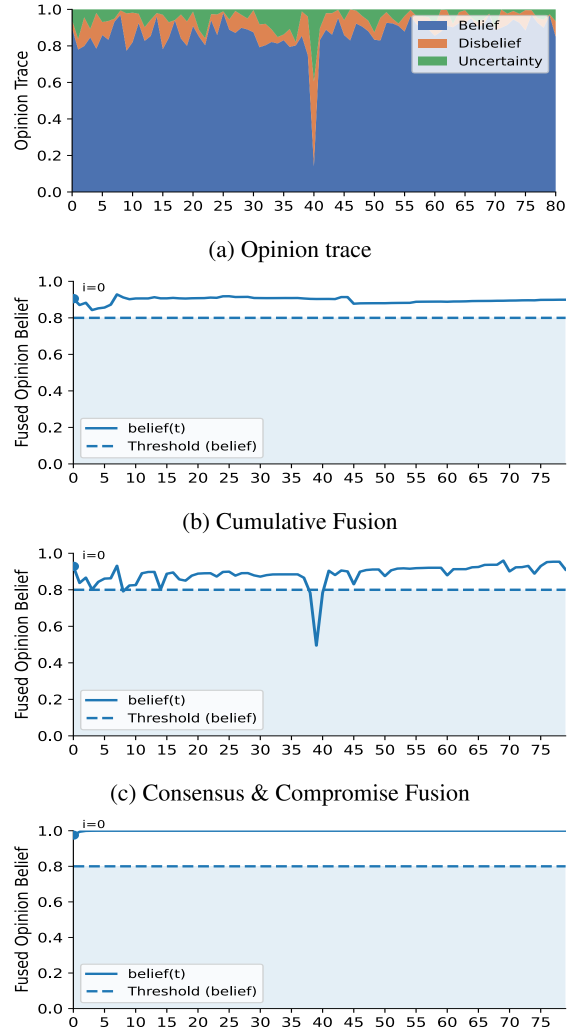
(a) Opinion trace

(b) Cumulative Fusion

(c) Consensus & Compromise Fusion

(d) Constraint Fusion

Figure 1: Scenario 1: Opinion trace (a) and fused belief using different instantiations of fusion operators (b-d). Shaded regions indicate values below the decision threshold. First instance of threshold crossing labeled i.



(a) Opinion trace

(b) Cumulative Fusion

(c) Consensus & Compromise Fusion

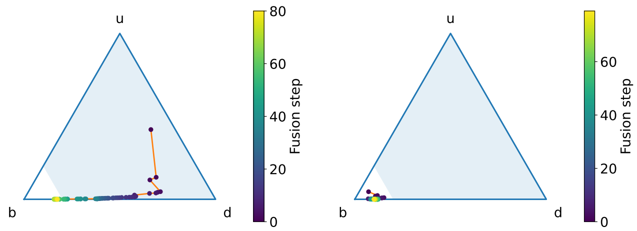
(d) Constraint Fusion

Figure 2: Scenario 2: Opinion trace (a) and fused belief using different instantiations of fusion operators (b-d). Shaded regions indicate values below the decision threshold. First instance of threshold crossing labeled i.

This illustrates that, even for the same opinion trace over $safe$, different choices of fusion operator induce different dynamics and verdicts for $\mathbf{G}\varphi$. In practice, this allows the global safety property to be tuned to different levels of criticality: more tolerant operators (such as cumulative fusion) model robustness to brief misclassifications, whereas more conservative operators (such as consensus & compromise in this setting) treat the same disturbance as a decisive counterexample to persistent safety.

Geometric interpretation of temporal satisfaction across scenarios To consolidate the scenario-specific results and provide an intuitive geometric interpretation of temporal satisfaction, we now visualise opinion traces directly in opinion space. This perspective abstracts from individual time-series plots and highlights how different trace dynamics give rise to satisfaction of eventuality and global properties. We thus visualize the satisfaction of temporal operators over an *opinion simplex*, i.e., the geometric space of admissible opinion states (e.g., belief masses) whose components are non-negative and sum to one. Each point on the simplex represents an opinion, and a trace induces a *trajectory* through this space over time. In Fig. 3, the shaded region marks simplex states that *violate* θ_b (opinions below threshold), while the unshaded region corresponds to states that *satisfy* θ_b.

The operator $\mathbf{G}\varphi$ requires that θ_b holds at *every* time step along the trajectory: the trace satisfies $\mathbf{G}\varphi$ iff the trajectory never enters the shaded region of the opinion simplex. This is illustrated in Fig. 3b, where the opinion remains above the threshold throughout the entire trace. In contrast, $\mathbf{F}\varphi$ re-



(a) Trace where $\mathbf{F}\varphi$ is satisfied. (b) Trace where $\mathbf{G}\varphi$ satisfied.

Figure 3: Opinion trajectory across trace demonstrating interpretation of $\mathbf{G}\varphi$ and $\mathbf{F}\varphi$ on the opinion simplex. Shaded region indicates opinions which violate threshold.

quires that φ holds at *some* time step: the trace satisfies $\mathbf{F}\varphi$ iff the trajectory reaches the unshaded region at least once. The right side of Fig. 3a shows this case: the trajectory initially lies in the shaded region but later crosses into the unshaded region, thereby meeting $\mathbf{F}\varphi$ while failing $\mathbf{G}\varphi$.

Related Work

Reasoning about agents and their beliefs has been a central topic in AI and, in particular, in the multi-agent systems community for several decades (Fagin and Halpern 1987; Wooldridge 2003). Classical work often builds on epistemic, doxastic, and BDI-style logics (Van Ditmarsch, van Der Hoek, and Kooi 2008; van der Hoek and Wooldridge 2012), where beliefs and knowledge are modelled as modal operators over Boolean propositions and verified via model checking over Kripke-style structures (Wooldridge et al. 2002; Bordini et al. 2006; Lomuscio, Qu, and Raimondi 2017). These approaches motivate the explicit specification of agent beliefs, but they are typically based on formal system models and exhaustive verification over all possible executions, rather than representing belief as an explicitly uncertain internal state and *monitoring* the runtime evolution of such states along a single observed execution trace.

A second line of work handles uncertainty in temporal reasoning by enriching either the logic or the verification process. Probabilistic temporal and epistemic logics such as PCTL and PLTL (Konur 2010) treat probability as part of the specification, with semantics over the distribution of all paths of an underlying probabilistic model. As probabilistic extensions to multiagent doxastic logics, APT logic (Shakarian et al. 2011) and its extension PDT logic (Martiny and Möller 2016) represent beliefs as (possibly interval-valued) probabilities over sets of possible temporal evolutions, assign temporal rules a frequency-based semantics via frequency functions, and are primarily aimed at design-time reasoning tasks. Related generalized annotated and fuzzy temporal logics, as implemented in PyReason (Aditya et al. 2023), support offline reasoning about fuzzy or interval-valued truth of propositions in a single (possibly graph-structured) model.

The area of runtime verification under uncertainty (Taleb, Hallé, and Khoury 2023) considers incomplete, imprecise, or out-of-order traces, and abstraction-based, language-based, or statistical techniques are employed to compute conservative, multi-valued, or probabilistic verdicts for Boolean temporal properties. In runtime approaches, uncertainty is attached to events, timestamps, or unobserved system states and to the resulting verdicts over traces, rather than to a structured, evolving internal belief state of an agent.

SLTL differs from these lines of work by considering SL opinions as atomic propositions and giving temporal operators an evidential semantics. In contrast to model checking of modal logics and to existing (probabilistic or quantitative) runtime monitors, SLTL directly specifies and monitors constraints on the belief trajectory of a knowledge-grounded, *uncertainty-aware agent*, rather than on Boolean abstractions of its perception or on external world states.

Discussion and Conclusion

In this paper, we have introduced Subjective LTL (SLTL), an uncertainty-aware, temporal specification language whose atomic propositions are SL opinions about predicates in an agent's symbolic knowledge model. With its evidential semantics, SLTL aims to enable richer analyses of temporal properties than standard Boolean LTL-based evaluation of thresholded traces under a purely 'logical' interpretation.

Interpreting SLTL formulae as constraints on *belief trajectories* makes the framework suitable for knowledge-grounded semantic agents. In such agents, atomic predicates refer to ground-truth concepts in the real world, while their truth is inferred from potentially noisy perception outputs. SLTL allows these beliefs to remain explicitly uncertain and constrains their temporal evolution under evidential fusion. This supports several agent-centric use cases:

- *Runtime assurance:* agent beliefs are monitored, an alert is created when confidence in a safety-relevant invariant drops, and appropriate mitigation actions are triggered.
- *Introspective reasoning:* agents use the monitor to track evidence about their own modules (e.g. "sensor-reliable") and apply SLTL properties to ensure they do not act on the basis of poorly supported beliefs.
- *Explainability:* SLTL monitoring exposes why a property is (not) asserted, e.g., whether violation is due to strong negative evidence or merely lack of positive evidence.

In this way, SLTL can serve as a hybrid (symbolic + evidential) building block in semantic agent architectures, complementing both symbolic reasoning and ML-based perception.

This paper has focused on establishing the conceptual framework and an illustrative example based on stylised synthetic opinion traces. Future work may include systematic comparisons of SL fusion operators with respect to monitoring sensitivity and robustness across application scenarios, as well as the integration of SLTL monitors into concrete agent architectures (e.g., in combination with neural perception modules) for runtime assurance and human-understandable explanations. Furthermore, applying SLTL to real system outputs is an important next step for empirical validation. Together, these directions aim to strengthen the practical applicability of the proposed approach and to support more transparent and reliable assurance of safety-critical AI systems.

Acknowledgments

The research leading to these results is funded by the German Federal Ministry for Economic Affairs and Energy within the project "Safe AI Engineering – Sicherheitsargumentation befähigendes AI Engineering über den gesamten Lebenszyklus einer KI-Funktion". We would also like to thank the reviewers for their helpful feedback.

References

Aditya, D.; Mukherji, K.; Balasubramanian, S.; Chaudhary, A.; and Shakarian, P. 2023. PyReason: Software for open world temporal logic. *arXiv preprint arXiv:2302.13482*.

Baier, C.; and Katoen, J.-P. 2008. *Principles of model checking*. MIT press.

Bordini, R. H.; Fisher, M.; Visser, W.; and Wooldridge, M. 2006. Verifying multi-agent programs by model checking. *Autonomous agents and multi-agent systems*, 12(2): 239–256.

Fagin, R.; and Halpern, J. Y. 1987. Belief, awareness, and limited reasoning. *Artificial intelligence*, 34(1): 39–76.

Jøsang, A. 2016. *Subjective logic*, volume 3. Springer.

Konur, S. 2010. Real-time and probabilistic temporal logics: An overview. *arXiv preprint arXiv:1005.3200*.

Leucker, M.; and Schallhart, C. 2009. A brief account of runtime verification. *The journal of logic and algebraic programming*, 78(5): 293–303.

Lomuscio, A.; Qu, H.; and Raimondi, F. 2017. MCMAS: an open-source model checker for the verification of multi-agent systems. *International Journal on Software Tools for Technology Transfer*, 19(1): 9–30.

Manna, Z.; and Pnueli, A. 2012. *Temporal verification of reactive systems: safety*. Springer Science & Business Media.

Martiny, K.; and Möller, R. 2016. PDT logic: a probabilistic doxastic temporal logic for reasoning about beliefs in multi-agent systems. *Journal of Artificial Intelligence Research*, 57: 39–112.

Shakarian, P.; Parker, A.; Simari, G.; and Subrahmanian, V. V. 2011. Annotated probabilistic temporal logic. *ACM Transactions on Computational Logic (TOCL)*, 12(2): 1–44.

Taleb, R.; Hallé, S.; and Khoury, R. 2023. Uncertainty in runtime verification: A survey. *Computer Science Review*, 50: 100594.

van der Hoek, W.; and Wooldridge, M. 2012. Logics for multiagent systems. *Ai Magazine*, 33(3): 92–92.

Van Ditmarsch, H.; van Der Hoek, W.; and Kooi, B. 2008. *Dynamic epistemic logic*. Springer.

Wooldridge, M. 2003. *Reasoning about rational agents*. MIT press.

Wooldridge, M.; Fisher, M.; Huget, M.-P.; and Parsons, S. 2002. Model checking multi-agent systems with MABLE. In *Proceedings of the 1st Int. Joint Conf. on Autonomous Agents and Multiagent Systems: part 2*, 952–959.