Cognition Envelopes for Bounded AI Reasoning in Autonomous UAS Operations

Pedro Antonio Alarcón
Granadeno
University of Notre Dame
Computer Science and Engineering
Notre Dame, IN, USA
palarcon@nd.edu

Arturo Miguel Bernal Russell
University of Notre Dame
Computer Science and Engineering
Notre Dame, IN, USA
arussel8@nd.edu

Sofia Nelson
University of Notre Dame
Computer Science and Engineering
Notre Dame, IN, USA
snelso24@nd.edu

Demetrius Hernandez
University of Notre Dame
Computer Science and Engineering
Notre Dame, IN, USA
dhernan7@nd.edu

Maureen Petterson
mpetters@nd.edu
Computer Science and Engineering
Notre Dame
Notre Dame, Indiana, USA

Michael Murphy
mmurph51@nd.edu
Computer Science and Engineering
Notre Dame
Notre Dame, Indiana, USA

Walter J. Scheirer
University of Notre Dame
Computer Science and Engineering
Notre Dame, IN, USA
walter.scheirer@nd.edu

Jane Cleland-Huang
University of Notre Dame
Computer Science and Engineering
Notre Dame, IN, USA
janehuang@nd.edu

Abstract

Cyber-physical systems increasingly rely on Foundational Models such as Large Language Models (LLMs) and Vision-Language Models (VLMs) to increase autonomy through enhanced perception, inference, and planning. However, these models also introduce new types of errors, such as hallucinations, overgeneralizations, and context misalignments, resulting in incorrect and flawed decisions. To address this, we introduce the concept of Cognition Envelopes, designed to establish reasoning boundaries that constrain AI-generated decisions while complementing the use of meta-cognition and traditional safety envelopes. As with safety envelopes, Cognition Envelopes require practical guidelines and systematic processes for their definition, validation, and assurance. In this paper we introduce the concept of a Cognition Envelope within the life-critical domain of small autonomous Uncrewed Aerial Systems deployed on Search and Rescue missions. We describe an LLM/VLM-supported pipeline for dynamic clue analysis and a Cognition Envelope based on probabilistic reasoning and resource analysis. We experimentally evaluate the approach, comparing the correctness of decisions made by our Clue Analysis Pipeline, with and without Cognition Envelope oversight. Finally, we synthesize lessons learned and identify an initial set of key software engineering challenges for systematically designing, implementing, and validating Cognition Envelopes for AI-supported decisions in cyber-physical systems.

Keywords

cognition envelope, foundational models, LLM, VLM, autonomous decisions, probabilistic lost person models, small uncrewed aerial systems, search and rescue

1 Motivation

Advances in Large Language Models (LLMs) and Vision Language Models (VLMs) have laid the foundations for improved perception, inferencing, and planning for autonomous Cyber-Physical Agents, such as small Uncrewed Aerial Systems (sUAS) [14, 20]. At the same time, the growing autonomy enabled by these models heightens the risk that AI errors, such as hallucinations, result in incorrect understanding and flawed decision-making [21, 49]. In domains such as aerial Search and Rescue (SAR), where autonomous sUAS are deployed in life-critical operations, such failures can compromise mission objectives, endanger human lives, and erode trust. To address these challenges, we introduce *Cognition Envelopes*, designed to detect flawed decisions that occur when reasoning outcomes contradict available evidence, violate operational constraints, or lack internal justification due to model errors or hallucinations. Cognition Envelopes serve as independent, external reasoners that employ diverse heuristic, probabilistic, and symbolic logic approaches to assess the soundness of an LLM/VLM-generated plan. As illustrated in Figure 1, they differ from meta-cognition, which operates within the reasoning process itself, when a model self-critiques and refines its own plans [4, 39]. They also differ from safety envelopes, which assure physical and operational safety by constraining system behavior within predefined limits [5]. In contrast, Cognition Envelopes regulate the outcomes of the reasoning process to prevent, or minimize, unsound or unjustified decisions. However, while safety envelopes are supported by a mature ecosystem of standards and assurance frameworks [17–19, 25, 26, 33, 34, 36, 37, 45], no such guidance currently exists for defining, validating, or assuring *Cognition Envelopes*. This paper therefore introduces the concept of *Cognition Envelopes*, illustrates their practical use for AI-supported reasoning in the domain of Search and Rescue (SAR) with small Uncrewed Aerial Systems (sUAS), and lays out preliminary guidance
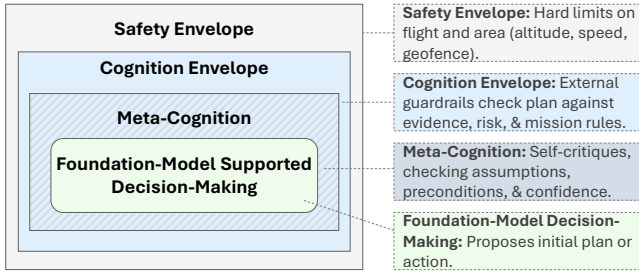
Figure 1: Layered decision-making guardrails: the outer Safety Envelope sets hard limits, while the Cognition Envelope applies external guardrails on inner-level decisions.

and open software engineering challenges related to integrating them in software intensive systems.

The discussion of Cognition Envelopes is timely tiven the growing body of work in the robotics space, where researchers have proposed the use of LLM and VLM-based techniques for diverse tasks such as image interpretation, intelligent task assignment, and vision-based trajectory planning [2, 8, 22, 40, 53]. However, this research area of AI autonomy has progressed with little to no focus on associated software engineering practices and challenges.

To describe and explore the use of Cognition Envelopes, we start by presenting a concrete application within the sUAS SAR domain where a *Clue Analysis Pipeline* (CAP) is used to manage visual clues, such as discarded items or footprints, that are detected by sUAS during a search. The CAP utilizes multi-modal foundational models to analyze the captured image of a clue, determine its relevance, and plan a subsequent action. We integrate two safeguards into our Cognition Envelope. The first is a Probability-Based SAR Model (*pSAR*) that computes the probability of a lost person being found in any given area of the search region at a given stage of an unfolding mission. The second is a simple Mission Cost Evaluator (MCE) that examines the cost of executing a search plan. Together, the pSAR and the MCE are responsible for assessing, and potentially constraining plans, generated by the CAP. Our worked example from the SAR domain, not only illustrates the use of a Cognition Envelope in practice, but also serves as a vehicle for exploring the Software Engineering challenges involved in developing such systems, including requirements elicitation, architectural design, validation, and edge-based deployment under resource constraints.

The remainder of this paper is organized as follows. Section 2 presents the LLM/VLM pipeline that supports clue analysis. Sections 3 and 4 describe and evaluate the use of a Cognition Envelope, including an assessment of its generalizability. Section 5 outlines a set of open Software Engineering challenges, and Sections 6–8 discuss related work, threats to validity, and conclusions.

2   Clue Analysis using Foundational Models

For purposes of this paper we focus on one aspect of SAR, namely the detection of a clue by the sUAS and its subsequent analysis using foundational AI models. Figure 2 illustrates this scenario, showing two sUAS ready for dispatch on a mission, where they ultimately discover a discarded backpack, serving as a critical clue. The approach described in this paper, is agnostic as to whether

(a) Two hexacopters are dispatched on SAR missions to search for a missing person.

(b) A backpack is found at a trailhead during a search for a lost person.

Figure 2: An sUAS discovers a backpack at the trailhead while conducting a trail-based search for a lost hiker. This clue could trigger mission-level adaptations.

the sUAS has been dispatched on a search by a human or by an automated planner. At the time that a clue is detected, we assume that a mission is ongoing and that sUAS have been enacting search-based tasks assigned either by human operators or by an automated mission planner (e.g., [10]). In this scenario, each sUAS is equipped with onboard Computer Vision (CV) using the YOLO-World Model for open-vocabulary detection tasks [3]. When a potential clue is detected, it is geolocated using a terrain model [35], and a representative image frame of the clue is selected and passed to the CAP.

The terrain model used for clue geolocation purposes and also by the CAP pipeline, is constructed using multi-source data fusion, combining satellite imagery with public geospatial datasets. These include Digital Terrain Maps (DTMs) for elevation [42], hydrographic features (e.g., lakes, streams) [47], forest ecosystems [48], and road/trail networks [46]. Satellite imagery is acquired via commercial APIs and processed by a convolutional neural network (CNN) trained to classify terrain categories into diverse features. CNN and USGS labels are cross-referenced to resolve conflicts and fill gaps, deferring to USGS where reliable and to CNN where USGS data are missing or outdated. The resulting terrain is discretized into a uniform grid of cells, each representing the dominant terrain type at that location. The cells are automatically organized into labeled terrain clusters to produce homogeneous regions such as woodlands or waterways. Large clusters are further divided into smaller spatial partitions that form compact units suitable for mission planning and trajectory generation, and that serve as named search areas within the pipeline's task-planning process.

2.1   AI-Supported Clue Analysis Pipeline

The CAP pipeline consists of four primary stages responsible for interpreting and captioning an image of a clue, assessing its relevance, planning appropriate tasks, and deciding whether a human should vet the generated plan. Each stage of the CAP pipeline is LLM-enabled utilizing gpt 4.o, and includes a set of inputs, a carefully engineered prompt, and a set of outputs. In addition, all stages, except the first, use Retrieval Augmented Generation (RAG) to provide guidance that is integrated into the prompt. For illustrative
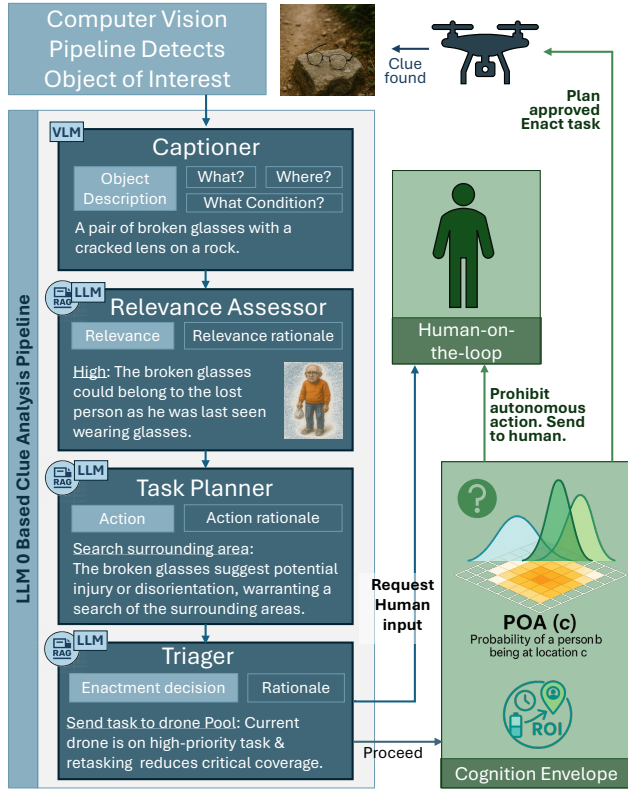
Figure 3: The Clue Analysis Pipeline includes four stages. The VLM generates a caption for the clue, while RAG + LLM analyzes its relevance. The Task Planner and Triager then decide whether to enact, revise, or defer the resulting action to a human operator. The Cognition Envelope approves the task or redirects it to a human for further assessment.

purposes we describe these stages using the example depicted in Figure 3. Additional information, including complete prompts, is provided as supplemental material.

- **Pipeline Trigger:** The pipeline is triggered when the sUAS sends a geolocated image frame depicting a candidate clue to the input node of the pipeline.

- **Stage 1 - Captioner:** The Captioner receives the clue as a single image frame and is prompted to generate a structured description. The interaction involves the following inputs (green) and outputs (blue) for the clue of the broken glasses depicted in Figure 3. Colored text is shownverbatim from text used within the prompt and from the generated responses.

> **Image:** A single image frame
>
> **Clue:** A pair of broken glasses with a cracked lens on a rock.
> **What?** A pair of broken glasses
> **Where?** On a rock
> **Condition?** Cracked lens

The captioner outputs a structured description of the clue for use in subsequent pipeline steps.

- **Stage 2 - Relevance Checker:** The relevance checker, as the name suggests, is responsible for assessing the relevance of the clue with respect to the lost person. It accepts the original description of a lost person, as well as the description of the clue output by the Captioner (Stage 1). RAG is used to retrieve up to five associated pieces of guidance related to relevance, and these are integrated into the prompt. For example, *"A clue is more likely to be relevant if it closely matches the known clothing or belongings of the lost person"*. The LLM evaluates the lost-person and clue descriptions, using the guidance where helpful, to classify the clue's relevance. It produces a categorical ranking (Very High, High, Medium, Low, None) together with a rationale. Inputs and outputs are shown below:

> **Person:** An elderly man wearing glasses, an orange sweater, blue plaid pants, and gym shoes.
> **Clue:** A pair of broken glasses with a cracked lens on a rock.
>
> **Relevance:** High
> **Rationale:** The broken glasses match the description of the lost man wearing glasses and were found beside a dirt path, suggesting possible alignment with his travel route.

- **Stage 3 - Task Planner:** The task planner is responsible for planning the action to be taken based on the clue. Actions are defined as search tasks, each task focused on searching a specific named area of the terrain model. This includes both boundaries of areas (e.g., shorelines of a river) or internal areas (e.g., a section of a forest or lake). This stage takes three inputs describing the clue relevance, its rationale (output from Stage 2), and a description of terrain features in the vicinity of the clue. These include the terrain cluster in which the clue was found (on), immediate neighbors (adjacent), and all other clusters within a predefined radius (nearby). It also retrieves five pieces of relevant advice about task-planning from the RAG dataset, and this is considered when generating an output action. As a result it outputs a prioritized list of possible actions, three of which are illustrated here:

> **Relevance:** High
> **Rationale:** The broken glasses match the description of the lost man wearing glasses and were found beside a dirt path, suggesting possible alignment with his travel route.
> **Location:** ON: Name of terrain feature where clue was found
> ADJACENT: List of immediately adjacent terrain features
> NEARBY: List of additional features within distance D
>
> **Actions:** Trail-10, Trail-11, Lake-5

The task planner relies on an underlying terrain model to interpret the context of the clue and to generate appropriate actions. Although the terrain model serves a variety of additional purposes across the SAR system, we introduce it here in relation to the task planner.

- **Stage 4: Triager** Finally, Stage 4 is responsible for determining how the plan will be enacted. Current options are limited to (a) permitting the current sUAS (i.e., the sUAS that found the clue) to perform the task, (b) sending the task to the drone pool for prioritization alongside other tasks to be performed, or (c) requesting an operator to vet the decision and to potentially

override it. This stage accepts five inputs comprising the clue description, clue relevance, weather conditions, drone swarm status (i.e., location, health, and priority levels of all drones), LKP and Elapsed Time (ET). Once again, the LLM prompt is enriched with five pieces of relevant advice concerning triaging from the RAG dataset and outputs the triage decision.

> **Clue:** A pair of broken glasses with a cracked lens on a rock.
> **Relevance:** High
> **Weather:** Light=Bright, Weather=Snow, Temp=Hot
> **Drones:** RED:30mins/High; BLUE:10 mins/Med; AQUA:35mins/Med
> **Sighting:** LKP: 64.328122, -20.516289, ET: 60 mins
>
> **Assignment:** Send task to Drone Pool

- **Post-Pipeline Enactment** When a task is assigned to a specific sUAS or to the sUAS pool, the assignee dynamically adapts from its current task to the new task. If the decision is made to engage humans, the sUAS continues its current action while a request for further analysis is made to a human operator, via an active Graphical User Interface.

While the CAP serves a critical purpose in interpreting clues and dynamically generating meaningful action plans, as previously discussed, its use of foundational models means that it can potentially generate erroneous results, and therefore its decisions need to be checked by a runtime Cognition Envelope. In our case example, the Cognition Envelope is comprised of the pSAR and the heuristic-based MCE utility.

3 Establishing a Cognition Envelope

Establishing a Cognition Envelope involves three key steps of requirements scoping, solution design, and validation. In this section we focus on scoping and design. Validation is deferred to Section 4, which describes how experimentation can be used within the software testing process to assess the efficacy of a Cognition Envelope.

3.1 Scoping Envelope Responsibilities

We start by establishing the scope of the Cognition Envelope to establish clearly defined guardrails around the CAP with the joint goals of achieving trustworthy and reliable autonomous decision-making. A Cognition Envelope can be designed in two distinct ways. In a black-box configuration, it inspects only the final outcome of an LLM-guided process; while in a white-box configuration it looks inside the pipeline, examining the outputs of intermediate reasoning stages such as captioning, relevance assessment, and action selection in order to detect flaws or inconsistencies before they influence the final output. We opted to treat the pipeline as a black box for two reasons. First, individual stages already incorporate localized meta-cognitive checks that support self-consistency, and second, developing cognition envelopes for each component would add significant design complexity and runtime cost. We capture these decisions through a high-level quality-related goal, three functional requirements (FR-1, FR-2, FR-3), and one non-functional requirement (NFR-4) depicted in Figure 4. Responsibilities for satisfying the three requirements are distributed across the pSAR, MCE, and human operators. The pSAR is charged with checking that decisions to search various areas of the terrain align with current
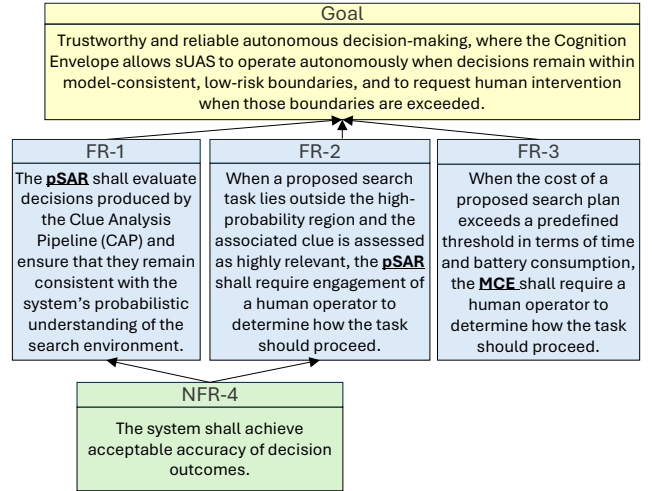

Figure 4: Decomposition of Cognition Envelope requirements showing requirements related to probabilistic coherence (pSAR) and Mission Cost Evaluator (MCE)

.

probabilities, while the MCE checks for, and curbs decisions that are costly in terms of time and power consumption. Notably, the MCE is simple to implement and validate, while the pSAR is more challenging. We therefore focus our validation process on the pSAR.

3.2 Designing pSAR as a Cognition Envelope

To design the pSAR as a Cognition Envelope, we followed a systematic design process, exploring alternative architectural options and evaluating their ability to satisfy the stated requirements [29]. The final design establishes pSAR, using a probability-based potential-field approach, which is well suited to modeling established SAR best practices and for reasoning under uncertainty [16]. Probability-based potential fields model spatial likelihood as a continuous surface in which higher values represent regions with greater expected relevance to the search, allowing the system to reason about search priorities through gradient-based inference rather than discrete Bayesian updates. However, A full mathematical exposition of this model lies beyond the scope of this paper; instead, we summarize its key principles and its role within the Cognition Envelope.

3.2.1 Probability of Area for Lost Person. pSAR builds on the concept of Probability of Area (POA) to quantify the likelihood of the lost person's presence throughout the search region. Given a last known point (LKP), elapsed time since establishing the LKP, a person profile, and environmental data, the pSAR models the likelihood of the lost person being in any particular area of the search region. Acting in its role within the Cognition Envelope, pSAR then evaluates how closely a CAP-generated plan aligns with these probabilities. The underlying distributions arise from two primary concepts of *reachability* and *affinity*.

- *Reachability:* Reachability is modeled as a spatial reasoning problem in which each terrain cell is associated with a relative ease of traversal derived from underlying environmental features. Rather than relying on discrete Bayesian updates, pSAR uses a continuous

probability-based potential field $R(c)$ to represent how reachable different regions are over time. The approach draws upon studies documenting walking rates in diverse terrains, and the impact of slope and water [23, 44]. It encodes the expected effort required to reach any location from the last known position and decays smoothly with increasing distance or terrain difficulty. Computed from terrain-weighted traversal times, $R(c)$ provides an efficient and interpretable way to reason about likely movement corridors.

• *Affinity Kernel:* Affinity captures how strongly a lost person's movement is drawn toward or aligned with salient environmental features such as roads, trails, or shorelines. The *Affinity Field $A(c)$* is implemented using smooth *radial basis functions (RBFs)* that assign high affinity to nearby features and gradually decrease influence with distance. pSAR currently defines RBF fields for 11 types of features, including categories corresponding to terrain clusters and sub-clusters for roads, waterways and their shorelines, woodlands and their boundaries, buildings, and open areas. Each feature is associated with an affinity strength parameter, and the total affinity for cell c is the product of its feature-specific affinities. The resulting affinity surface $A(c)$ provides a probabilistic representation of environmental preference based on the lost-person profile.

The probability of the lost person being in cell c is represented by the probability-of-area $p(c)$ computed as the product of physical reachability $R(c)$ and behavioral affinity $A(c)$, combined into a unified spatial estimate of where the person could plausibly be. This fine-grained cell is then aggregated into spatially compact subclusters using k-means partitioning. These subclusters serve as the primitive units for planning search-based tasks.

3.2.2 Evidence Updates. After a clue has been detected with sufficient confidence, the POA is dynamically updated to reflect the new evidence. Each update adjusts spatial belief according to the type of information conveyed by the clue. For example, clues that confirm the victim's past presence at a specific location refine probabilities locally around that point, while directional or feature-based clues reshape belief along implied movement paths or terrain structures. Finally, negative findings, such as area searches that are completed without finding a clue or the lost person, suppress probability in those areas. The adjustment process balances prior beliefs against the strength and reliability of the new evidence, ensuring that confident, well-supported clues meaningfully redirect the search while uncertain or low-value observations exert proportionally smaller influence. In this way, the model continuously integrates evolving field intelligence into a coherent, up-to-date spatial estimate of likely victim location. Notably, when the CAP detects a clue and classifies it with medium relevance or higher, the POA evidence is updated, and the updated evidence is used in the Cognition Envelope to check the plans generated by the CAP.

3.3 Evaluating CAP Plans with pSAR

We evaluate each CAP-generated search plan against the model's probability-based ranking using two complementary signals: (a) the candidate's **percentile rank** and (b) its **ratio to the top-ranked candidate**. The percentile rank indicates a search area's relative position within the sorted list (1.0 = top, 0.0 = bottom), while the

ratio-to-top quantifies magnitude agreement as

$$\rho = \frac{q(\kappa_{\text{suggested}})}{q(\kappa_{\text{top}})}, \quad (1)$$

where $q(\cdot)$ is the normalized candidate score. These metrics convey orthogonal information: percentile captures relative ordering, whereas ratio-to-top detects cases where candidates are similarly ranked yet substantially weaker in absolute terms. Relying on percentile alone can be misleading in flat distributions, while the ratio-to-top ensures the proposed candidate is not orders of magnitude weaker than the best option.

To quantify model uncertainty, we compute normalized Shannon entropy over the candidate distribution:

$$H_{\text{norm}} = -\frac{1}{\log |\mathcal{K}|} \sum_{\kappa \in \mathcal{K}} q(\kappa) \log q(\kappa) \quad (2)$$

where 0 indicates a sharply peaked distribution and 1 reflects complete uncertainty. This entropy score governs how strictly or loosely we evaluate CAP plans, balancing exploitation under confidence with exploration under uncertainty.

Each candidate κ receives a decision value $D(\kappa)$ corresponding to one of three possible outcomes: **ACCEPT**, **ALERT**, or **REJECT**, determined by whether it satisfies entropy-adaptive thresholds on percentile rank r and ratio-to-top ρ.

$$D(\kappa) = \begin{cases} \text{ACCEPT} & \text{if } r \geq r_{\text{accept}}(H) \text{ and } \rho \geq \rho_{\text{accept}}(H) \\ \text{ALERT} & \text{if } r \geq r_{\text{alert}}(H) \text{ or } \rho \geq \rho_{\text{alert}}(H) \\ \text{REJECT} & \text{otherwise} \end{cases} \quad (3)$$

The thresholds $r_{\text{accept}}(H)$, $\rho_{\text{accept}}(H)$, $r_{\text{alert}}(H)$, and $\rho_{\text{alert}}(H)$ adapt to entropy through linear interpolation between strict (low H) and loose (high H) regimes:

$$r_{\text{accept}}(H) = r_{\text{accept}}^{\text{strict}} + H \cdot (r_{\text{accept}}^{\text{loose}} - r_{\text{accept}}^{\text{strict}}) \quad (4)$$

and similarly for $\rho_{\text{accept}}(H)$, $r_{\text{alert}}(H)$, and $\rho_{\text{alert}}(H)$.

The **ALERT** band captures borderline cases that warrant human review: a candidate triggers an alert if it meets either the percentile or ratio threshold. When the model is confident, we apply stricter alert thresholds near $r_{\text{alert}}^{\text{strict}}$ and $\rho_{\text{alert}}^{\text{strict}}$, filtering out weak suggestions before they reach operators. As entropy rises and the probability field flattens, the thresholds shift toward $r_{\text{alert}}^{\text{loose}}$ and $\rho_{\text{alert}}^{\text{loose}}$, flagging more candidates for review and increasing human involvement under higher uncertainty.

4 Validating Cognition Envelopes

Validating the effectiveness of a Cognition Envelope capability, especially a probabilistic one such pSAR, is non-trivial because its performance depends not only on algorithmic correctness but also on how well it captures operational constraints, mission dynamics, and less tangible concepts such as human expectations. Traditional unit tests are insufficient for such systems, as correctness cannot be assessed solely through input–output verification. Instead, validation must rely on large-scale simulations that capture environmental variability, operational constraints, and the interaction between autonomous reasoning and human oversight. We therefore set up the validation process as an experiment guided by the following research question:

Vignette 2: Washington

Search Region:

(46.822778,-120.376389)-(46.768333, -120.234444)

Environment: Clear weather, daylight, cold temperatures with sloped terrain with many trails and streams.
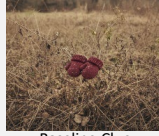
Person Description: A three year old little girl, wearing a red sweater, a red warm hat, and a red pair of mittens, brown pants and brown winter shoes.

LKP: 46.800556, -120.288056
ET: 180 minutes

Clue Location: 46.800000, -120.287222
Clue Context: Surrounded by low-vegetation with a stream nearby

Baseline Clue

Modified Clue

Non-relevant Clue

Figure 5: A Vignette establishes the context for each SAR contextualized decision point within our experiments. The vignette includes a lost person profile, environment, and clue data such as an image frame and the clue location. This example shows baseline information used for Test 2 and additional clues used in its variants.

Does the pSAR probabilistic Cognition Envelope provide adequate safeguards against inappropriate decisions made by foundational LLM and VLM models within the Clue Analysis Pipeline?

4.1 Experimental Design for Validating pSAR

We structured the experiment around concrete snapshots of decision points referred to as *vignettes*.

4.1.1 Vignette Design Process. Three members of the research team identified five regions, each of which had been the scene of a previous real-world SAR event. For each region they created two distinct *vignettes* describing the lost person, weather conditions, and a found clue. As illustrated in Figure 5, each vignette encapsulates information about the lost person, the search region, the current state of the sUAS swarm, and a discovered clue. The lost-person profile describes relevant characteristics and capabilities (e.g., an avid hiker) and the elapsed time (ET) since the last known point (LKP). The region description captures search boundaries, terrain, weather conditions, and lighting. The sUAS data specifies the number, location,

and health (e.g., battery level) of available aircraft. Finally, each clue is represented by a single image frame depicting the detected object, and the geographical coordinates at which it was found.

The first vignette for each region was anchored around a documented SAR report of a lost person; however, given sparse public information, we added additional data such as clothing descriptions and weather information. We also carefully inspected the terrain data and any known coordinates associated with the historical account of the search, and then used this information to place a clue at a meaningful. The second vignette used the same general region of terrain but but was constructed around a hypothetical scenario based on known behaviors of lost persons. While real-world clues are varied in nature, including examples such as footprints, crushed grass, cell-phone beeps, lost clothing or dropped water bottles, we focused our experiments upon objects such as clothing or other personal items that are within the scope of current CV models and VLM. We leave other forms of clue detection, such as footprints and crushed grass, to future work.

Each test suite included a total of seven unique tests. In addition to the baseline test ($V0$), six additional variants were created for each vignette as follows: $V1$ included a distorted version of the original clue (e.g., blurred, obfuscated, or damaged), while $V2$ included a non-relevant clue (e.g., a child's toy when searching for an adult hiker). $V3$ represented modifications to sUAS-related or mission parameters, while $V4$ included environmental and weather changes from the baseline scenario. For all of these four variants the clue was placed at a location that was within expected range of the search given the LKP and elapsed time. This meant that the pSAR would likely compute a non zero POA value for the terrain features around the clue's location. In contrast, for variants $V5$ and $V6$ the clue was placed at a remote location from the LKP, where it had a higher likelihood of being out of the predicted search area. For all six variants we experimentally modified *elapsed time* (ET), running tests at ET = 10, 20, 40, 60, and 90 minutes for each case. As *reachability* is impacted by elapsed time, the boundaries of the POA frontiers tend to spread out with higher ET levels, causing increased POA levels around the clues location over time. Conversely at low levels of ET, some clues were expected to have negligible POA levels.

We provide examples of the baseline test for each of the five regions, and provide the complete set of test cases as supplemental documents (see https://tinyurl.com/cain2026-cog-env). None of these test cases had been used in the initial design of the CAP or the pSAR, and all tests and their variants were constructed by two members of the team who had not developed either of these techniques.

- **Rock River, Illinois** [Test 1, V0]
  Search region: (41.470261, -90.531366) to (41.446453, -90.403751). *Lost person:* A teenage boy wearing a red shirt, red bucket hat, brown shorts, sneakers, and glasses. *Clue found:* A red hat floating in the water. *News report:* https://tinyurl.com/sar-rockriver

- **Kittitas, Washington** [Test 2, V0]
  Search region: (46.822778, -120.376389) to (46.768333, -120.234444). *Lost person:* A three year old little girl, wearing a red sweater, a red warm hat, and a red pair of mittens, brown pants and brown

winter shoes. *Clue found:* A pair of red mittens on the ground. *News report:* https://tinyurl.com/sar-washington

- **Mesa County, Colorado** [Test 3, V0]
  Search region: (38.962223, -108.333285) to (38.956922, -108.314291). *Lost person:* A moderately experienced hiker wearing a purple jacket, purple hat with orange goggles, black pants, hiking boots, a large black backpack, and carrying two hiking sticks. *Clue found:* Smoke rising above the trees. *News report:* https://tinyurl.com/sar-mesa

- **Pulaski, Arkansas** [Test 4, V0]
  Search region: (34.590254, -92.266513) to (34.552362, -92.189868). *Lost person:* Elijah, a 16-year-old male wearing a woolly orange-brown hat, orange sweater, dark blue–yellow striped pants, and gym shoes. Last seen arriving at a church on a purple bicycle. *Clue found:* A brownish knit hat on the gravel. https://tinyurl.com/sar-arkansas

- **Los Angeles, California** [Test 5, V0]
  Search region: (34.265727, -118.112756) to (34.217358, -118.048056). *Lost person:* An elderly, experienced hiker and camper wearing a purple jacket, purple winter hat with black and orange goggles, black pants, hiking boots, a large black backpack, and carrying two hiking sticks. *Clue found:* A pair of goggles on sandy colored rock. *News report:* https://tinyurl.com/sar-california

Additionally for experimental purposes, for each unique clue location we queried the terrain model and retrieved the set of terrain clusters surrounding that area. They included (1) the terrain cluster on which the clue was placed, (2) the immediately adjacent clusters, and (3) additional clusters within a predefined radius of 10 cells. An example showing the clusters for the baseline clue in Figure 5 is shown below, indicating that a large vegetation region (Low_Vegetation-42) was decomposed into smaller subsections, and that a stream was also nearby.

On:	Low_Vegetation-42-edge-82
Adjacent to:	Low_Vegetation-42-edge-26, Low_Vegetation-42-edge-71, Low_Vegetation-42-edge-7, Low_Vegetation-42-edge-28, Low_Vegetation-42-edge-145, Low_Vegetation-42-edge-163
Nearby:	Low_Vegetation-42-edge-80, Low_Vegetation-42-edge-82, Stream_River-7-edge-3

Notably for future real-world deployments, this data would be retrieved dynamically by sUAS at the time that a clue is discovered.

4.2 Experimental Infrastructure

We implemented the Clue Analysis Pipeline using GPT-4.0 accessed through the OpenAI API, with inference performed at a temperature of 0.2 to ensure consistent outputs. The system was built with the LangChain framework (langchain, langchain-community, langchain-openai) and used a FAISS vector store (faiss-cpu) for retrieval-augmented grounding. pSAR was implemented in Python via numpy and standard libraries.

To execute each test (or test variant), we first ran the CAP using clues and other data defined in each test case. CAP outputs were recorded in a JSON file. This part of the experiment was conducted on a system with an Intel Core Ultra 7 256V (2.20 GHz) processor, 16 GB RAM, and a 64-bit Windows operating system, with an average
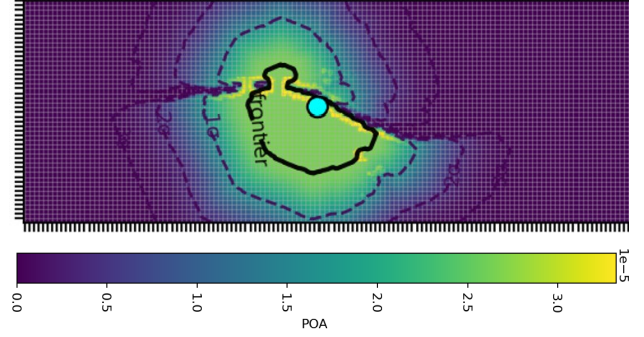

Figure 6: Visualization of the Probability of Area (POA) from the LKP (aqua circle). Probabilities are computed based on reachability and affinity and influenced by elapsed time since establishing the Last Known Point. This example shows where a river impedes reachability, but other types of impedances such as cliffs or marshland can have similar effects.

runtime of 12 seconds per clue. The pSAR part of the experiment was run on a server with 26 Intel Xeon processors, 159GB RAM, and an Ubuntu 22.04 operating system. The execution time of pSAR is highly dependent on the overall size of the region, ranging from an average runtime of 3 seconds for Region 3 ($\sim$ 1 square km) to an average runtime of 1 minute for Region 2 ($\sim$ 66 square km). The input to pSAR was provided as a json file containing the general scenario information as well as the output from CAP. Additionally, a second json file was given with the terrain data. pSAR initially processes the terrain data to build a scene, incorporating the priors of the scenario to build the initial POA map. When given a valid clue location, pSAR updates its probabilities to incorporate this information. Finally, the results are saved in an output json file containing the guardrail decisions for all CAP candidates and clues.

4.3 Results and Analysis

Following execution of the experiments, we first examined whether the CAP appropriately categorized the clues as *relevant* or *non-relevant* using the previously described rubric. Each test suite included one variant ($V2$) that included a non-relevant clue. For example in Figure 5, the vignette included an old boot which was clearly not relevant to the lost child. Results from the experiment showed that in every case the non-relevant clue was successfully rejected. Further, in eight out of the ten test suites, all relevant clues were marked as relevant. However, in one case the CAP failed to mark a clue as relevant due to severe occlusion (caused by a bird flying in-front of the clue in the image), and in another case the CAP failed to understand the significance of campfire smoke. However, this problem could be remedied in future implementations through adding additional guidance to the RAG dataset. Overall Stages 1 and 2 of the CAP pipeline achieved 95% accuracy with 47 true positives, 0 false positives, 10 true negatives, and 3 false negatives. The remainder of our discussion focuses on the cases where clues were marked as relevant and search plans were generated and forwarded to the pSAR for evaluation.

Next, we evaluated the pSAR performance on evaluating search plans generated by the CAP. To perform this evaluation we classified test cases into two categories. Group 1 included those for which the clue was placed within the vicinity of the currently active search region (Tests variants V0, V1, V3, V4), while Group 2 included those for which clues were placed at more extreme coordinates and were potentially outside the active search region (Variants V5, V6). We evaluated the pSAR performance under two distinct scenarios. First, when it evaluated search plans using existing POA scores without updating the scores based on the location and potential relevance of the found clue, and second, after pSAR updated its probabilities to reflect the discovered clue. We ran a total of 360 unique tests. Of these 240 tests belonged to Group 1 and 120 to Group 2.

Results from this experiment are summarized in Figures 7 and 8. Figure 7 shows results obtained when the pSAR did not update the underlying probability model based on the clue. They show that in Group 1, 53% of plans were approved, 43% were directed to human operators for review, and only 5% were rejected. This contrasted with Group 2, which included 26% accepts, 30% alerts, and 44% rejects. An inspection of individual cases showed that the three major inhibitors were distance from the LKP, Elapsed Time (which varied under the different treatments), and terrain barriers such as rivers or cliffs, all of which impacted current POA computations, and in turn the pSAR's decisions. In contrast, Figure 8 reports results when pSAR updated its underlying models to reflect the discovered clue prior to analyzing the planned search tasks. This had a strong effect of increasing probabilities around the area of the clue, thereby increasing the approval rate of search plans in the vicinity of the clue for all test cases, whether in Group 1 or Group 2. An initial inspection of the outcomes indicated that most non-approval decisions were caused when the clue was placed at a location that the pSAR deemed as unlikely to have been reachable given the LKP and elapsed time.

These results suggest that it is important to update the pSAR model when a clue is discovered, as this increases the probability of a person being in areas that surround the clue or are naturally reachable from the clue's location. Updating the model leads to higher approval ratings for clue-related plans generated by the CAP, and potentially increased levels of autonomy. Given high degrees of approval when the model is updated, the MCE element of the Cognition Envelope could also play an important role here, by checking that the time and power consumption needed by the sUAS to perform the task falls within acceptable thresholds. This is especially important, if the sUAS detects a clue in the distance, rather than close to its current position, as the the cost of flying to a distant location is potentially high. Although not part of our experiments, the MCE could cause some Approve decisions in Figure 8 to transition to Human engagement alerts.

These experiments serve as a proof-of-concept use of Cognition Envelopes in a viable sUAS application setting. The multi-faceted approach to the Cognition Envelope has demonstrated that the pSAR, potentially in partnership with the MCE and other services, is able to provide meaningful checks on decisions formulated by foundational models. At the same time, this is a new area of research, and many questions remained unanswere. We therefore lay these out in the following section.
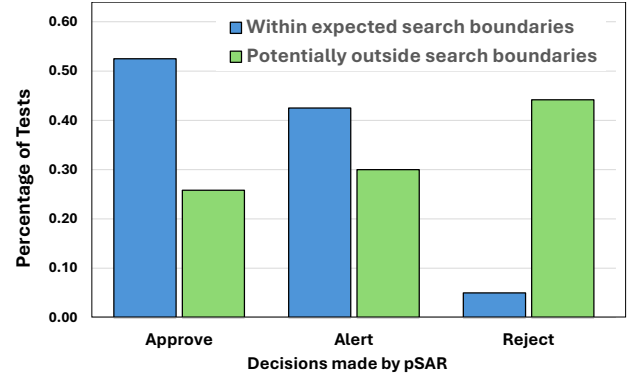

Figure 7: pSAR Decisions when using POA scores without updating based on the clue. In this case the majority of decisions within the current search area are approved, while many outside the area are rejected despite the presence of the clue.
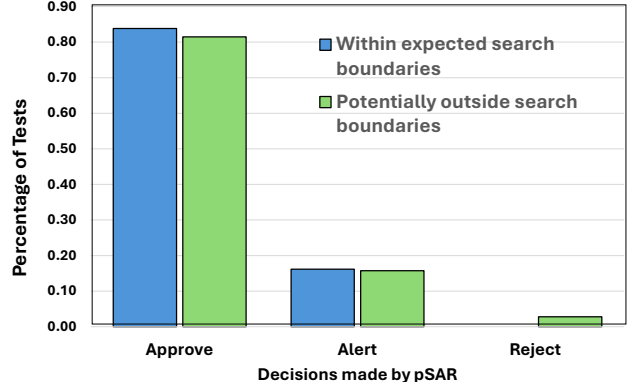

Figure 8: pSAR decisions made after the POA is updated, approving additional decisions, potentially increasing autonomy.

5 Open Challenges for Cognition Envelopes

This paper has presented the notion of Cognition Envelopes as a means for independently checking whether LLM/VLM reasoners produce valid plans. We have further demonstrated the effectiveness of a Cognition Envelope based on a probabilistic approach encapsulating uncertainty and resource consumption. In this section we discuss generalizability of the approach and open challenges for Software Engineering of Cognition Envelopes.

5.1 Generalizability of Cognition Envelopes

Cognition Envelopes are designed to provide an independent and orthogonal means of checking the soundness of LLM-generated decisions. They can be implemented through several complementary techniques, including probabilistic reasoning or statistical models to assess evidence alignment and uncertainty; symbolic or rule-based logic that defines constraints and permissible structures; knowledge consistency checks that ensure LLM outputs conform to established rules and facts; ROI or utility thresholding that authorizes actions

only when the expected mission value justifies the associated cost and risk; and human-engagement triggers that transfer control to an operator when confidence or supporting evidence falls below acceptable bounds. In this paper we have demonstrated and validated the use of a single Cognition Envelope in one application domain. While it is out of scope for this paper to fully answer the generalizability question, we briefly discuss their potential use across three additional systems.

In our first example, Yaqoot et al. used an LLM to parse an operator's description of a simple mission, and a VLM to analyze an image of the scene and to generate generate waypoints the UAV should take to reach a target location [52]. As with our LLM pipeline, probabilistic reasoning and ROI (utility) thresholding would make a particularly effective Cognition Envelope, because the former validates that proposed waypoints are consistent with mission evidence and belief state, while the latter approves execution only when the expected information gain justifies the time, energy, and risk.

In a second example, Emami et al. [9] deployed an LLM-enabled In-Context Learning (ICL) framework for public-safety UAVs that performed mission tasks such as path planning, velocity control, and data-collection scheduling. Their approach embedded goals, rules, and environmental constraints directly into structured prompts, enabling the model to reason from examples without retraining. While this improved adaptability and reduced latency, constraint checking occurred within the model itself rather than through an external Cognition Envelope. Conceptually, a Cognition Envelope could complement their approach by verifying decisions against mission evidence, assessing confidence and ROI before task execution, ensuring consistency between proposed paths and objectives, and engaging human operators whenever confidence or safety fell below acceptable limits.

Our third example extends beyond sUAS to the broader Cyber-Physical Systems domain. Xie et al. [50] used an LLM to reason about unseen regions and propose efficient exploration paths for UAV navigation. In this scenario, we conceptualized a two-tier Cognition Envelope: symbolic logic defined admissible flight corridors and prevented illegal maneuvers, while probabilistic reasoning estimated information gain and updated confidence over time. Knowledge-consistency checks, ROI thresholding, and human-engagement triggers together ensured that the LLM's proposals remained safe, efficient, and aligned with mission constraints.

While these are merely examples, they indicate that Cognition Envelopes are potentially applicable across a broad-range of LLM-supported CPS. Clearly further work is needed to deploye and validate their use.

5.2 Open Challenges

Our experience in engineering a Cognition Envelope for the sUAS application, highlighted several Software Engineering challenges that we highlight below. This list represents an initial attempt to document the challenges and is not intended to be complete.

- **Scoping the Cognition Envelope**: A cognition envelope may be composed of multiple interacting techniques, that work together to establish boundaries around the reasoning process. Defining the exact role of a Cognition Envelope and designing a

trustworthy and reliable solution is challenging. Without clear scoping, the Cognition Envelope will fail to provide fundamental protections at the reasoning level and deliver a false sense of security. *Mitigation*: Establish a concise specification that lists the envelope's inputs, checks, decision roles, and outputs, and explicitly delineates what it verifies, what it vetoes, and what it merely monitors.

- **Ground-Truth Alignment under Uncertainty**: Cognition Envelopes depend on evidence from sensors, logs, or runtime models, but that data is often incomplete or noisy. Furthermore, that data may be shared by the LLM Envelope leading to confirmation bias in the results. *Mitigation:* Implement meta-monitoring to assess the reliability of the Cognition Envelope itself during runtime. This includes tracking the quality and coverage of underlying evidence sources, estimating uncertainty in belief updates and ROI evaluations, and flagging cases where the envelope's own confidence becomes unreliable. When self-confidence falls below a defined threshold, the system should report degraded reasoning fidelity, invoke human oversight, or revert to conservative operational modes. Over time, logs of these self-assessments can be used to calibrate and improve envelope reliability across missions.

- **Verifying the Verifier**: The Cognition Envelope's probabilistic rules, symbolic constraints, or logic can contain its own unique flaws or unintended interactions, meaning that faults in the logic of the Cognition Envelope could silently permit or block valid actions, undermining trust and reproducibility. *Mitigation:* Treat the Cognition Envelope as a first-class software component subject to the same assurance practices as the system it monitors. Rigorously test each rule or inference path, simulate mission scenarios to expose false approvals and rejections, and periodically audit envelope behavior against logged evidence. Considering including a lightweight mechanism to estimate and report the envelope's own confidence in its assessments.

- **Appropriate Human Engagement**: Designing effective hand-off criteria and interfaces is as difficult as developing the cognitive logic itself. The system must decide when and how to involve a human without overwhelming the operator or creating ambiguity about responsibility. Cognition Envelopes lose purpose if humans are either constantly interrupted or summoned too late. *Mitigation*: Define clear and measurable engagement criteria tied to confidence levels, uncertainty thresholds, or detected anomalies. Provide interfaces that explain why intervention is requested and what options are available. Use adaptive notification strategies that escalate from informative cues to mandatory hand-off only when confidence or safety margins fall below defined limits.

- **Explainability and Auditability:** Every approval, rejection, or hand-off should include a traceable rationale that engineers, auditors, and regulators can inspect. *Mitigation:* Design the Cognition Envelope to produce concise, structured reasoning records for every decision, including inputs, applied rules, confidence levels, and resulting actions.

- **Comparing Meta-Cognition versus Cognition Envelopes:** Finally, another open challenge looks at the costs and benefits of meta-cognition versus cognition envelopes and when each

of them should be used.–*Mitigation:* Develop patterns that define the contexts under which each should be used, and explore opportunities for hybrid applications.

These and other challenges related to adaptability, adversarial robustness, and more, permeate the entire software development life cycle, influencing how Cognition Envelopes are specified, designed, implemented and validated. Addressing them early in requirements and design stages can prevent costly rework later, while integrating continuous validation and audit mechanisms ensures sustained reliability throughout deployment and evolution. Collectively, these issues also define a research roadmap for future work in the area of Cognition Envelopes.

6   Related Work

Recent work demonstrates growing integration of LLMs and VLMs in autonomous UAV operations [14]. Recent surveys have documented the transformative potential of LLMs in enhancing UAV decision-making, perception, and planning capabilities [20, 43]. Several systems have demonstrated LLM-enabled autonomous behaviors: vision-language reasoning for target waypoint generation for search and rescue [52], natural language-driven navigation for object search [28, 30], and edge-based in-context learning [9]. These applications showcase LLMs' capacity to interpret complex operational contexts and generate adaptive plans, yet they have progressed largely without systematic reasoning safeguards.

Hallucinations remain a fundamental limitation of LLMs, arising from training incentives that reward confident guessing over acknowledging uncertainty [21]. In safety-critical systems such errors can lead to incorrect situational assessments or dangerous misinterpretations of environmental conditions [41, 49]. Despite their severity, these limitations have received limited attention in the UAV autonomy literature.

Runtime assurance has emerged as a promising approach for ensuring safe behavior in learning-enabled autonomous systems whose complete verification at design time remains infeasible [12, 38]. Complementary approaches translate natural language instruction into formally verifiable specifications, enabling constraint-based runtime enforcement [27, 51]. These approaches focus primarily on physical or operational safety rather than validating the semantic correctness or evidential grounding of AI-generated reasoning.

Probabilistic reasoning and Bayesian inference have long served as foundational tools for managing uncertainty in robotics. Recent work has applied Bayesian approaches to UAV decision-making under uncertainty [6, 13, 24]. Such approaches naturally accommodate uncertainty and provide confidence bounds that reflect operational risk, capabilities that are directly relevant to evaluating the soundness of LLM-generated plans.

Metacognition has received growing attention as a mechanism for improving LLM reliability [1, 4]. Self-reflection techniques enable models to critique their own outputs, identify potential errors, and iteratively refine responses [32]. However, empirical studies reveal metacognitive deficiencies: models fail to recognize their knowledge limitations, provide overconfident answers even when correct information is unavailable, and struggle with robust internal reasoning necessary for full metacognition [11, 15]. These findings suggest that internal self-critique alone may be insufficient for high-assurance applications, motivating the need for external validation mechanisms.

LLM guardrails represent a related but distinct approach to constraining model behavior. Guardrails typically enforce structural or content-based constraints on the inputs and outputs through rule-based filters, embedding-based classifiers, or LLM-as-a-judge mechanisms [7, 31].

7   Threats to Validity

The work in this paper has presented the concept of a Cognition Envelope. However, there are several threats to validity.

Construct Validity. While we have discussed the potential generalizability of the concept across several CPS applications, this paper focused on a single Cognition Envelope applied to a single LLM-based pipeline. Moreover, the Cognition Envelope we developed and validated included only two techniques based on probabilistic reasoning and rule-guided human engagement. Further work is needed to fully verify the approach and explore additional envelope types and domains.

Internal Validity. Our primary experiments focused on the use of decision-related vignettes rather than on running experiments in actual missions. This choice was made because running a realistic mission in our high-fidelity simulation environment takes from 15 minutes to over an hour, depending on where sUAS are dispatched and where clues or the victim are placed. As the focus of this paper was on decision checking, the vignettes allowed us to study a larger number of scenarios with variants and injected mutants, and therefore collect more data for empirical analysis.

External Validity. A potential threat to external validity stems from the domain specificity and limited evaluation environment of our study. The Cognition Envelope and its underlying pSAR model were developed and tested primarily within a controlled simulation framework, and their generalizability to broader CPS domains or to field-deployed systems has not yet been verified. While we have previously conducted mission-level simulations using an earlier Bayesian-inspired version of the Cognition Envelope, the current approach has not yet been fully integrated or validated in physical drone missions. In future work, we plan to deploy the complete system both within our high-fidelity simulation environment and on physical sUAS platforms that are already equipped for onboard reasoning and decision checking.

Conclusion Validity. Our experiments focused only on the use of the Cognition Envelope and did not perform a head-to-head comparison against a pipeline-level meta-cognition layer. Our hypothesis, which we have not yet evaluated, is that the meta-cognition solution would suffer from the very problems we aim to address, and for that reason, an external and orthogonal approach such as pSAR is essential for achieving reliable decision making. Future experiments are needed to systematically compare these two approaches across realistic mission contexts and reasoning tasks.

Despite these limitations, this work has proposed a novel approach, and laid out open research challenges, for developing, integrating, and validating Cognition Envelopes as a distinct layer of assurance for LLM-based decision-making in CPS.

8 Conclusions

This paper has proposed the use of Cognition Envelopes as an independent mechanism for checking the decisions produced by systems that employ foundational models such as LLMs and VLMs for reasoning, inference, and planning. Rather than guaranteeing complete correctness, Cognition Envelopes evaluate each decision against a set of well-scoped criteria that define acceptable reasoning boundaries. Within these boundaries, they can provide specific assurances, for example, in our pSAR model, that any decision falling outside probabilistic expectations of the model, are automatically redirected for human review. Our experimental results have demonstrated that this approach can effectively detect flawed or unsafe decisions while maintaining operational continuity, providing a practical step toward building more trustworthy, transparent, and accountable AI-enabled systems. Finally, the paper has outlined a set of open Software Engineering challenges that form an initial roadmap for advancing research and practice in the development, validation, and assurance of Cognition Envelopes.

We provide access to supplemental materials associated with this paper: https://tinyurl.com/cain2026-cog-env.

References

[1] Ahsan Bilal, Muhammad Ahmed Mohsin, Muhammad Umer, Muhammad Awais Khan Bangash, and Muhammad Ali Jamshed. 2025. Meta-thinking in llms via multi-agent reinforcement learning: A survey. *arXiv preprint arXiv:2504.14520* (2025).

[2] Hongqian Chen, Yun Tang, Antonios Tsourdos, and Weisi Guo. 2025. Contextualized Autonomous Drone Navigation Using LLMs Deployed in Edge-Cloud Computing. In *2025 International Conference on Machine Learning and Autonomous Systems (ICMLAS)*. 1373–1378. doi:10.1109/ICMLAS64557.2025.10967934

[3] Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. 2024. Yolo-World: Real-time open-vocabulary object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16901–16911.

[4] Brendan Conway-Smith and Robert L West. 2024. Toward autonomy: Metacognitive learning for enhanced AI performance. In *Proceedings of the AAAI Symposium Series*, Vol. 3. 545–546.

[5] Rogério de Lemos, David Garlan, Carlo Ghezzi, Holger Giese, Jesper Andersson, Marin Litoiu, Bradley Schmerl, Danny Weyns, Luciano Baresi, Nelly Bencomo, Yuriy Brun, Javier Camara, Radu Calinescu, Myra B. Cohen, Alessandra Gorla, Vincenzo Grassi, Lars Grunske, Paola Inverardi, Jean-Marc Jezequel, Sam Malek, Raffaela Mirandola, Marco Mori, Hausi A. Müller, Romain Rouvoy, Cecília M. F. Rubira, Eric Rutten, Mary Shaw, Giordano Tamburrelli, Gabriel Tamura, Norha M. Villegas, Thomas Vogel, and Franco Zambonelli. 2017. Software Engineering for Self-Adaptive Systems: Research Challenges in the Provision of Assurances. In *Software Engineering for Self-Adaptive Systems III. Assurances*, Rogério de Lemos, David Garlan, Carlo Ghezzi, and Holger Giese (Eds.). Springer International Publishing, Cham, 3–30.

[6] Ruohai Di, Xiaoguang Gao, Zhigao Guo, and Kaifang Wan. 2018. A threat assessment method for unmanned aerial vehicle based on bayesian networks under the condition of small data sets. *Mathematical Problems in Engineering* 2018, 1 (2018), 8484358.

[7] Yi Dong, Ronghui Mu, Gaojie Jin, Yi Qi, Jinwei Hu, Xingyu Zhao, Jie Meng, Wenjie Ruan, and Xiaowei Huang. 2024. Building guardrails for large language models. *arXiv preprint arXiv:2402.01822* (2024).

[8] Mohamed Elnoor, Kasun Weerakoon, Gershom Seneviratne, Ruiqi Xian, Tianrui Guan, Mohamed Khalid M Jaffar, Vignesh Rajagopal, and Dinesh Manocha. 2025. VLM-GroNav: Robot Navigation Using Physically Grounded Vision-Language Models in Outdoor Environments. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*. 2391–2398. doi:10.1109/ICRA55743.2025.11128264

[9] Yousef Emami, Hao Zhou, Miguel Gutierrez Gaitan, Kai Li, Luis Almeida, and Zhu Han. 2025. From Prompts to Protection: Large Language Model-Enabled In-Context Learning for Smart Public Safety UAV. *arXiv preprint arXiv:2506.02649* (2025).

[10] Pedro Antonio Alarcon Granadeno and Jane Cleland-Huang. 2025. Coverage Path Planning for Holonomic UAVs via Uniaxial-Feasible, Gap-Severity Guided Decomposition. arXiv:2505.08060 [cs.RO] https://arxiv.org/abs/2505.08060

[11] Maxime Griot, Coralie Hemptinne, Jean Vanderdonckt, and Demet Yuksel. 2025. Large language models lack essential metacognition for reliable medical reasoning. *Nature communications* 16, 1 (2025), 642.

[12] Kerianne L Hobbs, Mark L Mote, Matthew CL Abate, Samuel D Coogan, and Eric M Feron. 2023. Runtime assurance for safety-critical systems: An introduction to safety filtering approaches for complex control systems. *IEEE Control Systems Magazine* 43, 2 (2023), 28–65.

[13] Niamat Ullah Ibne Hossain, Nazmus Sakib, and Kannan Govindan. 2022. Assessing the performance of unmanned aerial vehicle for logistics and transportation leveraging the Bayesian network approach. *Expert Systems with Applications* 209 (2022), 118301.

[14] Yibiao Hu, You Zhou, Zhengqiang Zhu, Xi Yang, Han Zhang, Kun Bian, and Hong Han. 2025. LLVM-drone: A synergistic framework integrating large language models and vision models for visual tasks in unmanned aerial vehicles. *Knowledge-Based Systems* (2025), 114190.

[15] Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2023. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798* (2023).

[16] Yijun Huang, Hao Li, Yi Dai, Gehao Lu, and Minglei Duan. 2024. A 3D Path Planning Algorithm for UAVs Based on an Improved Artificial Potential Field and Bidirectional RRT*. *Drones* 8, 12 (2024). doi:10.3390/drones8120760

[17] International Electrotechnical Commission. 2010. IEC 61508: Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems. IEC Standard.

[18] International Organization for Standardization. 2018. ISO 26262: Road Vehicles — Functional Safety. ISO Standard.

[19] International Organization for Standardization. 2019. ISO 21448: Road Vehicles — Safety of the Intended Functionality (SOTIF). ISO Standard.

[20] Shumaila Javaid, Nasir Saeed, and Bin He. 2024. Large Language Models for UAVs: Current State and Pathways to the Future. (2024). arXiv:2405.01745 [cs.RO] https://arxiv.org/abs/2405.01745

[21] Adam Tauman Kalai, Ofir Nachum, Santosh S Vempala, and Edwin Zhang. 2025. Why language models hallucinate. *arXiv preprint arXiv:2509.04664* (2025).

[22] Abdul-Manan Khan, Ikram Ur Rehman, Nagham Saeed, Drishty Sobnath, Fatima Khan, and Muazzam Ali Khan Khattak. 2025. Context-Aware Autonomous Drone Navigation Using Large Language Models (LLMs). In *Proceedings of the AAAI Symposium Series*, Vol. 6. 102–107.

[23] Robert J Koester. 2008. Lost Person Behavior: A search and rescue guide on where to look - for land, air and water. dbS Productions LLC. Charlottesville, VA, August.

[24] Simon Kohaut, Benedikt Flade, Devendra Singh Dhami, Julian Eggert, and Kristian Kersting. 2023. Mission design for unmanned aerial vehicles using hybrid probabilistic logic programs. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 1506–1513.

[25] Nancy Leveson. 2011. *Engineering a Safer World: Systems Thinking Applied to Safety*. MIT Press, Cambridge, MA.

[26] Nancy Leveson and John Thomas. 2018. *STPA Handbook*. MIT Press, Cambridge, MA.

[27] Jason Xinyu Liu, Ankit Shah, George Konidaris, Stefanie Tellex, and David Paulius. 2024. Lang2ltl-2: Grounding spatiotemporal navigation commands using large language and vision-language models. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2325–2332.

[28] Shubo Liu, Hongsheng Zhang, Yuankai Qi, Peng Wang, Yanning Zhang, and Qi Wu. 2023. Aerialvln: Vision-and-language navigation for uavs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 15384–15394.

[29] Bashar Nuseibeh. 2001. Weaving Together Requirements and Architectures. *Computer* 34, 3 (2001), 115–119. doi:10.1109/2.910904

[30] Adam Pardyl, Dominik Matuszek, Mateusz Przebieracz, Marek Cygan, et al. 2025. FlySearch: Exploring how vision-language models explore. *arXiv preprint arXiv:2506.02896* (2025).

[31] Traian Rebedea, Razvan Dinu, Makesh Sreedhar, Christopher Parisien, and Jonathan Cohen. 2023. Nemo guardrails: A toolkit for controllable and safe llm applications with programmable rails. *arXiv preprint arXiv:2310.10501* (2023).

[32] Matthew Renze and Erhan Guven. 2024. Self-reflection in llm agents: Effects on problem-solving performance. *arXiv preprint arXiv:2405.06682* (2024).

[33] RTCA, Inc. 2011. DO-178C: Software Considerations in Airborne Systems and Equipment Certification. RTCA Standard.

[34] John Rushby. 2019. *Runtime Assurance for Safety-Critical Systems*. Technical Report NASA/CR-2019-220266. NASA.

[35] Arturo Miguel Russell Bernal, Maureen Petterson, Pedro Alarcon Granadeno, Michael Murphy, James Mason, and Jane Cleland-Huang. 2025. Validating Terrain Models in Digital Twins for Trustworthy sUAS Operations. In *Proceedings of the Conference on Engineering Digital Twins (EDT), co-located with MODELS 2025*. Grand Rapids, MI, USA. Regular paper, presented in person.

[36] SAE International. 2010. ARP4754A: Guidelines for Development of Civil Aircraft and Systems. SAE Aerospace Recommended Practice.

[37] Ivo Schaefer, Bojan Cukic, John Hatcliff, and Kamesh Namuduri. 2022. Runtime Assurance for Cyber-Physical Systems: A Survey. *Comput. Surveys* 55, 5 (2022),

1–38. doi:10.1145/3502289

[38] John D Schierman, Michael D DeVore, Nathan D Richards, and Matthew A Clark. 2020. Runtime assurance for autonomous aerospace systems. *Journal of Guidance, Control, and Dynamics* 43, 12 (2020), 2205–2217.

[39] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems* 36 (2023), 8634–8652.

[40] Daeun Song, Jing Liang, Amirreza Payandeh, Amir Hossain Raj, Xuesu Xiao, and Dinesh Manocha. 2025. VLM-Social-Nav: Socially Aware Robot Navigation Through Scoring Using Vision-Language Models. *IEEE Robotics and Automation Letters* 10, 1 (2025), 508–515. doi:10.1109/LRA.2024.3511409

[41] Aditya K Sood, Sherali Zeadally, and EenKee Hong. 2025. The Paradigm of Hallucinations in AI-driven cybersecurity systems: Understanding taxonomy, classification outcomes, and mitigations. *Computers and Electrical Engineering* 124 (2025), 110307.

[42] C.A. Thatcher, Vicki Lukas, and J.M. Stoker. 2020. *The 3D Elevation Program and Energy for the Nation.* Fact Sheet 2019–3051. U.S. Geological Survey. 2 pages. doi:10.3133/fs20193051

[43] Yonglin Tian, Fei Lin, Yiduo Li, Tengchao Zhang, Qiyao Zhang, Xuan Fu, Jun Huang, Xingyuan Dai, Yutong Wang, Chunwei Tian, et al. 2025. UAVs meet LLMs: Overviews and perspectives towards agentic low-altitude mobility. *Information Fusion* 122 (2025), 103158.

[44] Waldo Tobler. 1993. Three presentations on geographical analysis and modeling. (Technical Report 93-1). National Center for Geographic Information and Analysis (NCGIA), University of California, Santa Barbara..

[45] Underwriters Laboratories. 2020. UL 4600: Standard for Evaluation of Autonomous Products. UL Standard.

[46] U.S. Geological Survey. 2023. The National Map - Transportation Datasets. U.S. Geological Survey, National Geospatial Program. https://data.usgs.gov/datacatalog/data/USGS:ad3d631d-f51f-4b6a-91a3-e617d6a58b4e Vector data for roads, trails, and transportation features. Data available from 2014 to present. Accessed June 25, 2025.

[47] U.S. Geological Survey. 2023. NHDPlus High Resolution (NHDPlus HR). https://www.usgs.gov/national-hydrography/nhdplus-high-resolution Accessed June 25, 2025.

[48] U.S. Geological Survey, Earth Resources Observation and Science (EROS) Center. 2025. *Annual NLCD (National Land Cover Database) — The Next Generation of Land Cover Mapping.* Fact Sheet 2025–3001. U.S. Geological Survey. doi:10.3133/fs20253001 First release (Collection 1.0) covers land cover and change from 1985 to 2023 at 30m resolution across CONUS. Accessed June 25, 2025.

[49] Jue Wang. 2024. Hallucination Reduction and Optimization for Large Language Model-Based Autonomous Driving. *Symmetry* 16, 9 (2024), 1196.

[50] Quanting Xie, Tianyi Zhang, Kedi Xu, Matthew Johnson-Roberson, and Yonatan Bisk. 2024. Reasoning about the Unseen for Efficient Outdoor Object Navigation. arXiv:2309.10103 [cs.RO] https://arxiv.org/abs/2309.10103

[51] Ziyi Yang, Shreyas S Raman, Ankit Shah, and Stefanie Tellex. 2024. Plug in the safety chip: Enforcing constraints for llm-driven robot agents. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 14435–14442.

[52] Yasheerah Yaqoot, Muhammad Ahsan Mustafa, Oleg Sautenkov, Artem Lykov, Valerii Serpiva, and Dzmitry Tsetserukou. 2025. Uav-vlrr: Vision-language informed nmpc for rapid response in uav search and rescue. *arXiv preprint arXiv:2503.02465* (2025).

[53] Zezhong Zhang, Chenyu Hu, Sunwoh Lye, and Chen Lv. 2025. A VLM-Drone System for Indoor Navigation Assistance with Semantic Reasoning for the Visually Impaired. In *2025 IEEE/SICE International Symposium on System Integration (SII)*. 1260–1265. doi:10.1109/SII59315.2025.10871009